PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-344282
(43)Date of publication of application : 14.12.2001

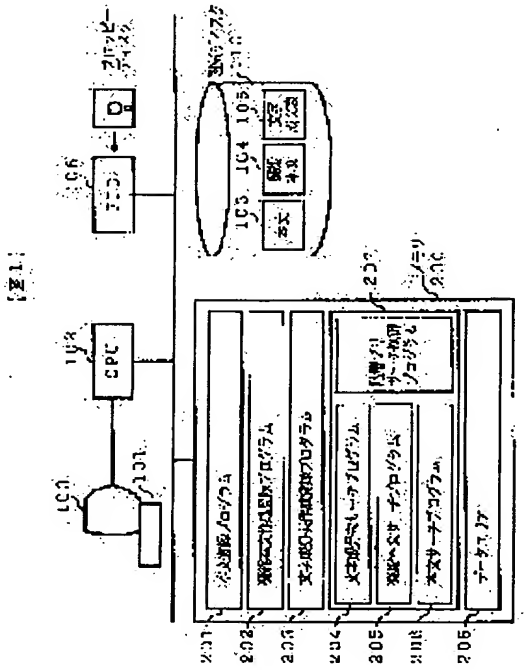
(51)Int.Cl. G06F 17/30

(21)Application number : 2001-064336
(22)Date of filing : 28.02.1991

(71)Applicant : HITACHI LTD
(72)Inventor : HATAKEYAMA ATSUSHI
FUJISAWA HIROMICHI
KATO KANJI
KAWAGUCHI HISAMITSU
MINEGISHI NAOKI

(54) METHOD AND DEVICE FOR DOCUMENT RETRIEVAL

(57)Abstract:
PROBLEM TO BE SOLVED: To solve such problem that the conventional full text search system requires much processing time because an entire text file is scanned from the beginning character by character and this system requires much time when being applied to a large- scale database.
SOLUTION: When a document is registered in a document database, a text character string of the registered document is divided for every character type such as Hiragana (cursive form of Japanese syllabary) or KANJI (Chinese character), and the inclusion relations between divided partial character strings are examined to generate a condensed text consisting of a set of partial character strings where character strings included in other character strings are excluded, and a character component table is generated where characters appearing in the condensed text are registered without duplication, and the condensed text and the character component table are registered in the document database besides the text of the registration object document. At the time of retrieval, the character component table is referred to extract a document including characters of a designated keyword, and next, the condensed text of the extracted document is referred to extract only documents corresponding to the condensed text including the partial character string of the designated keywords, and texts of extracted documents are referred to extract only a document meeting a retrieval condition given between keywords.



LEGAL STATUS

[Date of request for examination]	08.03.2001
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3303881
[Date of registration]	10.05.2002
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－344282
（P2001－344282A）
(43)公開日　平成13年12月14日(2001.12.14)

(51)Int.Cl.⁷	識別記号	ＦＩ	ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０６Ｆ　17/30	４１４	Ｇ０６Ｆ　17/30	４１４Ｂ　５Ｂ０７５
	１７０		１７０Ａ
	２２０		２２０Ａ

審査請求　有　　請求項の数４　ＯＬ　（全 27 頁）

(21)出願番号　特願2001－64336(P2001－64336)
(62)分割の表示　特願平3－58311の分割
(22)出願日　平成３年２月28日(1991.2.28)

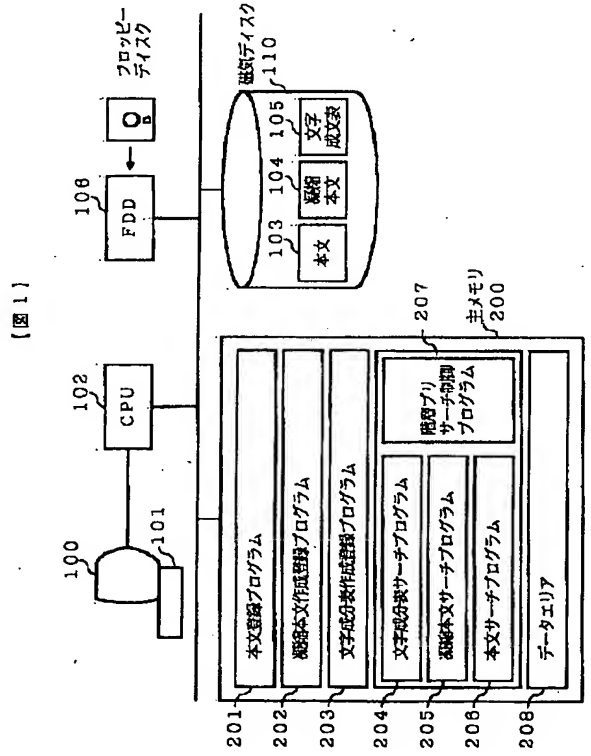
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　畠山　敦
東京都国分寺市東恋ケ窪１丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　藤澤　浩道
東京都国分寺市東恋ケ窪１丁目280番地
株式会社日立製作所中央研究所内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　文書検索方法および装置

(57)【要約】
【課題】　従来のフルテキストサーチ方式には，テキストファイル全体を先頭から一文字ずつ走査するために処理時間が掛かり，大規模なデータベースに適用すると時間が掛かるという課題があった。
【解決手段】　文書データベースに文書を登録する際，登録文書の本文文字列をひらがな，漢字等の文字種ごとに分割し，分割した各部分文字列の間で相互に文字列の包含関係を調べ，他の文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成し，凝縮本文中に現れる文字を重複なく登録した文字成分表を作成し，登録対象文書の本文のほかに凝縮本文と文字成分表を合わせて文書データベースへ登録する。検索時には指定キーワードの文字を含む文書を，文字成分表を参照して抽出し，次に抽出された文書の凝縮本文を参照して指定キーワードの部分文字列を含む凝縮本文に対応する文書のみを抽出し，該抽出文書の本文を参照してキーワード間に付与された検索条件を満たすもののみを抽出する。

【図１】



【特許請求の範囲】

【請求項１】文書情報を文字コードデータとして蓄積した文書データベースを対象として、検索者が指定したキーワードを含む文書を検索する文書検索装置において、該文書データベースに文書を登録する際、該登録文書の本文文字列をひらがな、漢字、及び英数字等の文字種ごとに分割し、分割した各部分文字列の間で相互に文字列の包含関係を調べ、他の文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成する手段と、該凝縮本文中に現れる文字を重複なく登録した文字成分表を作成する手段と、登録対象文書の本文のほかに凝縮本文と文字成分表を合わせて文書データベースへ格納する手段と、検索時に検索者が指定したキーワードを構成する全ての種類の文字を含む文書を、文字成分表を参照して抽出する文字成分表サーチ手段と、該文字成分表サーチで抽出された文書の件数を検査する手段と、該検査の結果件数が所定数以上の件数に達した場合は、凝縮本文を全件読み出して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出し、前記検査の結果件数が所定数以下の場合は、前記文字成分表サーチで抽出された文書の凝縮本文を参照して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出する凝縮本文サーチ手段と、抽出された文書の本文を参照して、キーワード間に付与された位置関係等の検索条件を満たすものを抽出する本文サーチ手段とを備えたことを特徴とする文書検索装置。

【請求項２】文書情報を文字コードデータとして蓄積した文書データベースを対象として、検索者が指定したキーワードを含む文書を検索する文書検索装置において、該文書データベースに文書を登録する際、該登録文書の本文文字列をひらがな、漢字、及び英数字等の文字種ごとに分割し、分割した各部分文字列の間で相互に文字列の包含関係を調べ、他の文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成する手段と、該凝縮本文中に現れる文字を重複なく登録した文字成分表を作成する手段と、登録対象文書の本文のほかに凝縮本文と文字成分表を合わせて文書データベースへ格納する手段と、検索時に検索者が指定したキーワードを構成する全ての種類の文字を含む文書を、文字成分表を参照して抽出する文字成分表サーチ手段と、該文字成分表サーチで抽出された文書の件数を検査する手段と、該検査の結果件数が一定数以上の件数に達した場合は、凝縮本文を全件読み出して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出する凝縮本文サーチ手段と、該凝縮本文サーチ手段により抽出された文書の本文を参照して、キーワード間に付与された位置関係等の検索条件を満たす文書を抽出し、前記の文字成分表サーチの結果件数が一定数以下の場合に該文字成分表サーチで抽出された文書に対する本文を参照して、指定キーワードを含むとともにキーワード間に付与された位置関係等の検索条件を満たす文書を抽出する本文サーチ手段とを備えたことを特徴とする文書検索装置。

【請求項３】文書情報を文字コードデータとして蓄積した文書データベースを対象として、検索者が指定したキーワードを含む文書を検索する文書検索方法において、該文書データベースに文書を登録する際、該登録文書の本文文字列をひらがな、漢字、及び英数字等の文字種ごとに分割し、分割した各部分文字列の間で相互に文字列の包含関係を調べ、他の文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成するステップと、該凝縮本文中に現れる文字を重複なく登録した文字成分表を作成するステップと、登録対象文書の本文のほかに凝縮本文と文字成分表を合わせて文書データベースへ格納するステップと、検索時に検索者が指定したキーワードを構成する全ての種類の文字を含む文書を、文字成分表を参照して抽出する文字成分表サーチステップと、該文字成分表サーチで抽出された文書の件数を検査するステップと、該検査の結果件数が所定数以上の件数に達した場合は、凝縮本文を全件読み出して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出し、前記検査の結果件数が所定数以下の場合は、前記文字成分表サーチで抽出された文書の凝縮本文を参照して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出する凝縮本文サーチステップと、抽出された文書の本文を参照して、キーワード間に付与された位置関係等の検索条件を満たすものを抽出する本文サーチステップとを有することを特徴とする文書検索方法。

【請求項４】文書情報を文字コードデータとして蓄積した文書データベースを対象として、検索者が指定したキーワードを含む文書を検索する文書検索方法において、該文書データベースに文書を登録する際、該登録文書の本文文字列をひらがな、漢字、及び英数字等の文字種ごとに分割し、分割した各部分文字列の間で相互に文字列の包含関係を調べ、他の文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成するステップと、該凝縮本文中に現れる文字を重複なく登録した文字成分表を作成するステップと、登録対象文書の本文のほかに凝縮本文と文字成分表を合わせて文書データベースへ格納するステップと、検索時に検索者が指定したキーワードを構成する全ての種類の文字を含む文書を、文字成分表を参照して抽出する文字成分表サーチステップと、該文字成分表サーチで抽出された文書の件数を検査するステップと、該検査の結果件数が一定数以上の件数に達した場合は、凝縮本文を全件読み出して、検索者が指定したキーワードを構成する部分文字列を含む凝縮本文に対応する文書を抽出する凝縮本文サーチステップと、該凝縮本文サーチステップにより抽出された文書の

本文を参照して、キーワード間に付与された位置関係等の検索条件を満たす文書を抽出し、前記の文字成分表サーチの結果件数が一定数以下の場合に該文字成分表サーチで抽出された文書に対する本文を参照して、指定キーワードを含むとともにキーワード間に付与された位置関係等の検索条件を満たす文書を抽出する本文サーチステップとを有することを特徴とする文書検索方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は，文書データベースを文字列を指定して文書の全文を対象として探索する検索技術に係わり，特に補助的なファイルを用いて高速な検索処理を実現するための検索技術に関する。

【０００２】

【従来の技術】従来の文書検索システムでは，登録する文書の内容を表す単語（キーワードと呼ぶ）をインデクスとする方式がとられている。しかし，この方式ではインデクサーとよばれるキーワード付けの専門家が文書を逐一読み，内容を理解した上で適切なキーワードを振る必要があった。この登録時の手間の掛かる作業を回避するために，「特開昭６３－１９８１２４」のような本文中に出現する単語を全てキーワードとしてインデクスファイルに登録する方法も提案されている。しかし，上記の方法ではインデクスファイルの作成時に，意味を持つ最小の単位の単語を決定するのが難しく，単語辞書あるいは，文法規則の不備のために，文章の解析に失敗して，重要な単語がキーワードとして抽出されないという問題がある。この問題を解決するために検索時に文書を文字コード化したテキストとして直接計算機に登録し，検索時にはテキストデータベース内の全ての文書の内容を読んで，与えられたキーワード（従来システムにおける統制キーワードと区別するために，以後検索タームと呼ぶ）を含む文書を探し出だすフルテキストサーチが提案されている。

【０００３】このフルテキストサーチ方式は，「情報処理学会研究報告　ｖｏｌ．８９，ｎｏ．６６　情報学基礎１４－７　テキストデータベース管理システムＳＩＧＭＡとその応用（１９８９．７．２７）」の第２節冒頭で述べられているように，テキストファイル全体を先頭から一文字ずつ走査することが大きな特徴である。こうすることにより，キーワードに対応する文書識別子等を記述したインデクスファイルがなくとも，テキストデータベースのテキスト本体を手掛かりに検索することが可能となる。すなわち，与えられた検索タームでテキストデータ全体を文字列探索し，検索タームが記述されている文書を検索結果として出力することができる。

【０００４】

【発明が解決しようとする課題】従来のフルテキストサーチ方式は，テキストファイル全体を先頭から一文字ずつ走査するために処理時間が掛かり，大規模なデータベースに適用すると時間が掛かるという課題があった。同文献第２節中にみられるように，汎用の大型計算機を持ってしても，２ＭＢ／ｓ程度の検索処理速度しか実現できない。この速度でも，数メガバイト程度のデータベースであれば，検索時間は実用域内に入る。しかし，オフィス等の実用規模のデータベースには数百メガバイトの容量が必要とされ，この場合には十分な検索レスポンスが得られないことになる。

【０００５】本発明の目的は，高速な文書検索方法および装置を提供することにある。

【０００６】

【課題を解決するための手段】上記課題を改善するために，以下の処理ステップから構成される文書検索方法を用い，該方法を実施する装置を構成する。

（１）本文自体を格納するステップ

（２）格納した本文を単語レベルで部分文字列へ分解し，分解した部分文字列間で相互に文字列の包含関係を調べ，他の部分文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成するステップ

（３）本文中で用いられている文字を重複なく集めた文字成分表を作成するステップ

（４）与えられた検索タームを文字レベルで分解し，検索タームを構成する全ての文字を含む文書を抽出する文字成分表サーチのステップ

（５）文字成分表で抽出された文書に対応する凝縮本文を参照し，与えられた検索タームを含む文書を抽出する凝縮本文サーチのステップ

（６）与えられた検索条件式が複数の検索ターム間の本文中での位置関係を指定している場合には，凝縮本文で抽出された文書に対応する本文データを参照し，与えられた検索タームを含み，なおかつ検索ターム間に付与された位置関係等の検索条件を満たすものを抽出する本文サーチのステップ

【作用】このように，文字成分表サーチ，凝縮本文サーチと階層的に絞り込みを行い最後に本文サーチを行う階層型プリサーチ手段を設けることによって，文字成分表サーチ，凝縮本文サーチで与えられた条件式を満たさない文書をテキスト本文を参照する以前に切り捨てて，検索対象のテキスト本文を探索する量を少なくすることができる。すなわち，検索処理時間に占める割合が高い本文検索処理時間を減らすことによって，全体の検索処理時間を短縮することが可能となる。例えば，「本文中に“画像”と“処理”とが同一の文（センテンス）内にある文書を探せ」という二つの検索タームの本文中での位置的な関係まで指定した条件式が与えられた場合，直接本文を参照する従来の方法では検索処理速度を２ＭＢ／ｓと仮定して，５００ＭＢのフルテキストを全て探索するのに２５０秒，すなわち約４分掛かる。しかし，階層型プリサーチでは，典型的な場合で，文字成分表でデータベース全件の１０％に，凝縮本文でさらにその１０％

に絞り込めたとすると，凝縮本文の容量が本文の３０％の場合，文字成分表の容量はデータベース全体からみると無視できるほど小さいので，検索すべき凝縮本文の容量は１５ＭＢで，本文データの探索量は全データベース量の１％，すなわち５ＭＢとなるため，２ＭＢ／ｓの検索速度でも，１０秒で検索処理を終了できることになる。このように，「階層型プリサーチ方式」では，「文字成分表」と「凝縮本文」という２段階のプリサーチを事前に行い，それぞれ「文字レベル」と「単語レベル」のふるいに掛け，最も時間を要する本文サーチの対象となる文書数をあらかじめ最小に絞り込んでおくことによって，探索文書容量を削減することができるため，等価的に非常に高速なフルテキストサーチが実現できることになる。また，条件式が単一の検索タームあるいは複数の検索タームでのＡＮＤ，ＯＲ，ＮＯＴ条件の場合には，凝縮本文サーチでの結果をそのまま最終検索結果とすることができる。なぜなら，凝縮本文中に存在している単語は，必ず本文中にも存在するためもう一度本文を検索する必要がないためである。このように，「単語レベル」での検索では処理時間のかかる本文サーチをまったく省略することができるため，より一層全体の検索処理時間を短縮することが可能となる。以上の処理ステップから構成されるフルテキストサーチ方法によれば，直接本文を探索する量を予め少なくすることができるため，高速なフルテキストサーチが可能となる。

【０００７】

【発明の実施の形態】以下，本発明の第一の実施例について図１を用いて説明する。本装置は，ディスプレイ１００，キーボード１０１，中央制御装置ＣＰＵ１０２，文字成分表１０５，凝縮本文１０４，及び本文１０３格納用ファイル１１０，フロッピディスクドライバ１０６，主メモリ２００から構成される。

【０００８】主メモリ２００には，本文登録プログラム２０１，凝縮本文作成登録プログラム２０２，文字成分表作成登録プログラム２０３，文字成分表サーチプログラム２０４，凝縮本文サーチプログラム２０５，本文サーチプログラム２０６，階層型プリサーチ制御プログラム２０７が格納されるとともに，データエリア２０８が確保されている。これらのプログラムはＣＰＵ１０２で実行される。文書の登録の際は，キーボード１０１から入力されるコマンドにより，ＣＰＵ１０２がフロッピディスクドライバ１０６に挿入されるフロッピディスクから文書データを読込み，本文登録プログラム２０１を実行して読み込んだ文書データを本文１０３としてファイル１１０へ格納する。次にＣＰＵ１０２は，凝縮本文作成登録プログラム２０２を実行して，本文１０３を単語レベルで部分文字列へ分割し，分割した部分文字列間で相互に文字列の包含関係を調べ，他の部分文字列に含まれる文字列を排除した部分文字列の集合からなる凝縮本文を作成し，これを凝縮本文１０４としてファイル１１０へ格納する。最後にＣＰＵ１０２は，文字成分表作成登録プログラム２０３を実行して，本文１０３中で用いられている文字を重複なく集めた文字成分表を作成し，これを文字成分表１０５としてファイル１１０へ格納する。検索の際には，キーボード１０１から入力された検索条件式が，ＣＰＵ１０２に送られる。ＣＰＵ１０２では，まず階層検索制御プログラム２０７を実行し，その制御に基づいて文字成分表サーチプログラム２０４，凝縮本文サーチプログラム２０５，本文サーチプログラム２０６を順次実行する。すなわち，文字成分表サーチでは，入力された検索条件式中の検索タームを文字レベルで分解し，検索タームを構成する全ての文字を含む文書を抽出する。そして，文字成分表で抽出された文書に対応する凝縮本文を参照し，与えられた検索タームを含む文書を抽出する。もし，与えられた検索条件式中に単一の検索タームか，もしくは複数の検索ターム間の論理的な関係が指定されているのみで，本文中での位置関係までは指定されていない場合には，ここで検索を終了し，凝縮本文サーチの結果を検索結果として出力する。それ以外の場合，すなわち与えられた検索条件式中に複数の検索ターム間の本文中での位置関係が指定されている場合には，凝縮本文サーチで抽出された文書に対応する本文データを参照し，与えられた検索タームを含み，なおかつ検索ターム間に付与された位置関係等の検索条件を満たすものを抽出し，検索結果として出力する。以上が本発明の第一の実施例のフルテキストサーチ装置の概略である。

【０００９】以下，本発明の特徴である文字成分表サーチ，凝縮本文サーチ，及び本文サーチと絞り込みを行う階層型プリサーチ方式の，登録及び検索方法について概略を説明する。まず「凝縮本文」と「文字成分表」の作成は，文書の登録時に自動的に行う。この処理内容を，図２に示す。本図で，登録すべき文書が入力されると，まずそのまま「本文」として格納する。

【００１０】次に，この「本文」から「凝縮本文」を作成する。「凝縮本文」は，「本文」の中から文字種（漢字、ひらがな、カタカナ、英字等）ごとに文字列を分割し，繰り返し現れる言葉の重複を排除して作成される。本文が「あいまい検索のための検索技術・・・・・」という文書１の場合には，「検索」が重複語として切り捨てられ，「あいまい」と「検索技術」及び「のための」が「凝縮本文」として残ることになる。また，「本文」から「文字成分表」を作成する。ここでは，「本文」に現われる文字を１ビットの情報で表す。文書１の例では，「あ」と「い」があるのでそれぞれ‘１’を，また「う」はないので‘０’を設定する。「検」と「索」も同様にそれぞれ‘１’を設定する。以下同様にして，文字成分表の該当文字部分に，「本文」にその文字がある場合には‘１’を，存在しない場合には‘０’を設定する。このようにして，文書の登録時に「凝縮本文」と

「文字成分表」を自動的に作成し，階層型プリサーチの準備をしておく。検索時には，図３に示すように，登録の逆の順序でこれらの補助ファイルを参照する。まず第１に，文字成分表サーチでは，文字成分表を参照し，検索ターム中の全ての文字に対応する文字成分表の該当文字部分に‘１’が設定されているものを選びだす。第２に，凝縮本文サーチでは，文字成分表で選びだされた文書の凝縮本文を参照し，条件式に与えられた検索タームを含む文書を選びだす。最後に本文サーチでは，検索タームの本文での出現位置が条件式と適合するものを選びだす。本図の例では，　検索［４Ｃ］理解すなわち，「“検索”と“理解”が本文中で４文字以内に近接して現れるものを探せ」という条件式で検索した例を示している。結果として文書４の“検索”と“理解”が本文中で４文字離れている文書が抽出される。

【００１１】以下，本実施例で用いる文字種分割・重複語排除型凝縮本文及び文字コード依存型文字成分表の作成方法と，これらを用いた階層型プリサーチの制御方法について具体的に説明する。まず最初に本実施例で用いる文字種分割・重複語排除型凝縮本文の作成方法について説明する。図４に示すように，まず本文テキストから文字種により文字列を分割する。この時の文字種とは，漢字，ひらがな，カタカナ，英字，数字，記号その他である。これらの単一文字種の連なりからなる文字列毎に本文の文字列を分割する。次に，分割した文字列のそれぞれについて，同一文書内にある他の部分文字列にその部分文字列がそっくり含まれてしまう場合，その文字列を重複文字列として凝縮本文の対象から排除する。例えば，“検索”という部分文字列は，同一文書内にある他の“知的検索技術”という部分文字列に完全に含まれるので，この“検索”は凝縮本文には登録しない。しかし，凝縮本文サーチでは，たとえ“検索”という文字列は凝縮本文に登録されていなくとも，“知的検索技術”の部分文字列としてヒットすることになる。

【００１２】このように，部分文字列の重複登録を排除して，得られた部分文字列には，図５に示すように文書毎に文字列の間にセパレータを挿入する。本図では，セパレータとして記号‘，’を用いている。図２，図３ではこのセパレータは記号‘｜’で表されているが，このセパレータは特に文字として表す必要はなく，文字に割り当てられていない特殊なコードを使用することもできる。

【００１３】次に本実施例で用いる文字コード依存型文字成分表の作成方法について説明する。図６に示すように，文字コード依存型文字成分表は，文字コードによって存在を示すビット情報として，１を立てるビット位置を決定する文字成分表である。本図ではシフトＪＩＳコードを例に説明している。同図で（ＸＸＸＸ）Ｈは文字コードを１６進表示したものである。例えば“検索”という文字列が文書１の本文中に存在することを示すのに，文書１のビットリストの（８Ｃ９Ｆ）Ｈ，（８ＤＦ５）Ｈ番目に１を設定する。ビットリスト中のこの文字に対応するビット位置を文字成分表のエントリ番号と呼ぶことにする。例えば‘検’のエントリ番号は（８Ｃ９Ｆ）Ｈ，または１０進表示すれば３５９９９となる。以上の文字成分表と凝縮本文を用いた階層型プリサーチの制御及びサーチ動作について説明する。まず検索条件式中の検索タームをそれぞれ一文字単位に分解し，文字成分表サーチを行う。ここでは与えられた検索タームを構成する文字コードに対応するビットリスト中のエントリ番号の位置がすべて１となるビットリストを持つ文書を求めることとなる。例えば，“検索”という文字列が検索タームとして与えられた場合，‘検’，‘索’に対応するビットリストの（８Ｃ９Ｆ）Ｈ，（８ＤＦ５）Ｈ番目のビットがすべて１である文書１，２，３，４，．．．を文字成分表サーチの検索結果とする。

【００１４】すなわち，図７に示すように‘検’を示す（８Ｃ９Ｆ）Ｈのエントリ番号のビットリスト７０１と，‘索’を示す（８ＤＦ５）Ｈのエントリ番号のビットリスト７０２との間でビット毎にＡＮＤ演算を施し，ビットＡＮＤ演算結果７０３を得る。このビットＡＮＤ演算結果７０３のビットリスト中で，１となっているビット位置に対応する文書番号が文字成分表サーチの検索結果としてのヒット文書を表すことになる。すなわち，‘検’と‘索’を全て含む文書が抽出されることになる。また，“湖”のように検索タームがただ１個の文字から構成される場合は，ここで文字成分表サーチの結果を出力して検索を終了することができる。次に文字成分表サーチで抽出された文書の凝縮本文に対してサーチを行う。ここでは図５のように文書毎に登録された凝縮本文の内容をスキャンして，与えられた検索タームを単語として含む文書を抽出する。つまり，‘検’と‘索’の2文字が“検索”と連続して現れる文書を抽出する。すなわち，‘検’と‘索’が含まれていても，“検出”と“探索”というように，別の単語として現われるようなものはここで切り捨ててしまう。このためには，文字成分表サーチで絞り込まれた文書毎の凝縮本文について本文テキストデータと同じように，一文字ずつスキャンしながら探索する。この時，文字成分表サーチで得られた結果の文書番号に対応する凝縮本文しかスキャンしない。例えば，文字成分表サーチの結果が文書番号１，２，３，４，．．．であれば，凝縮本文サーチでは，文書番号１，２，３，４，．．．の凝縮本文をスキャンする。そして，実際に凝縮本文中に検索タームが存在する文書を凝縮本文サーチの検索結果として出力する。このように，「階層型プリサーチ方式」では，「文字成分表」と「凝縮本文」という２段階のプリサーチを事前に行い，それぞれ「文字レベル」と「単語レベル」のふるいに掛け，最も時間を要する本文サーチの対象となる文書数をあらかじめ最小に絞り込んでおくことによって，

探索文書容量を削減することができるため，等価的に非常に高速なフルテキストサーチが実現できることになる。すなわち，文字成分表サーチでは，文字成分表が文字の存在を１ビットの情報で表しているため，サーチするデータ容量を極めて小さくすることができ，その結果検索時間も短時間に納めることが可能となる。さらに，キーワードを構成する文字毎のビットリストの論理積を取ることによって，キーワードに関連のない文書を大幅に切り捨て，以降の対象文書を格段に絞り込むことが可能となる。また，凝縮本文サーチでは，本文を直接スキャンするよりもデータ量が少ない分，検索処理時間が短縮できることになる。

【００１５】次に，本発明の第二の実施例を説明する。本実施例は，複数の検索タームが指定された場合でも，効率的に階層型プリサーチを行うことのできるフルテキストサーチ方法を提供するものである。例えば，「“検索”ＡＮＤ“理解”」という条件式が与えられたときには，まず，第１ステップとして文字成分表をサーチする。ここでは与えられた検索ターム毎にそのすべての文字を含む文書を探し，その後検索ターム間で与えられた条件を満たすような文書を出力する。「“検索”ＡＮＤ“理解”」という条件式の場合には，“検索”の２文字を含み，かつ“理解”の２文字を含む文書を探す。すなわち，「（‘検’ＡＮＤ‘索’）ＡＮＤ（‘理’ＡＮＤ‘解’）」従って，「‘検’ＡＮＤ‘索’ＡＮＤ‘理’ＡＮＤ‘解’」つまり，上記の４文字を同時に含む文書を検索する。次に，この文字成分表サーチの結果絞り込まれた文書の凝縮本文をサーチする。ここでは，指定されたキーワードが単語として現われる文書だけを抽出する。すなわち，“検索”と“理解”を両方同時に含む文書を検索する。この例の場合のように，検索ターム間の関係が“ＡＮＤ”，“ＯＲ”等の論理条件だけで，その他にキーワード間の位置関係を規定する条件が指定されていない場合には，ここで検索を終了し，凝縮本文サーチの結果を最終検索結果として出力する。もし，位置条件が指定されている場合には，凝縮本文サーチで抽出された文書の本文をサーチし，指定条件に合致するものを抽出し，これを最終検索条件として出力する。以上が本実施例における検索動作の説明である。このように，文字成分表サーチ，凝縮本文サーチで検索ターム間の論理積を取ることにより，複数個の検索タームが指定された場合でも，効率的に階層型プリサーチを行い，高速なフルテキストサーチを実現することができる。

【００１６】これより第三の実施例として，さらに一般的に階層型プリサーチの検索制御について詳細に説明する。図８にこのときの階層型プリサーチの制御の手順をＰＡＤ図にて説明する。ここでは「“計算機”と“知的インタフェース”のどちらかを含む文書を探せ」すなわち「“計算機”ＯＲ“知的インタフェース”」という検索式を例にあげて説明する。まず，最初にステップ８００で文字成分表サーチを行う。ここでは与えられた検索ターム毎にそのすべての文字を含む文書を探し，その後検索ターム間に与えられた複合条件を満たすような文書を出力する。この例では，図９に示すように“計算機”を構成する３個の文字のそれぞれについて文字成分表の該当するエントリ番号間のビットＡＮＤ演算を行い，次に同様に“知的インタフェース”を構成する９個の文字のそれぞれについて文字成分表の該当するエントリ番号間のビットＡＮＤ演算を行い，最後に先に作成した“計算機”に対するのときのビットＡＮＤ演算結果とそのビット列のＯＲ演算を行う。すなわち，「（‘計’ＡＮＤ‘算’ＡＮＤ‘機’）　ＯＲ（‘知’ＡＮＤ‘的’ＡＮＤ‘イ’ＡＮＤ　‘ン’ＡＮＤ‘タ’ＡＮＤ‘フ’ＡＮＤ　‘ェ’ＡＮＤ‘ー’ＡＮＤ‘ス’）」という検索式を実行することになる。これにより，“計算機”を構成する３個の文字をすべて含む文書，もしくは“知的インタフェース”を構成する９個の文字をすべて含む文書が抽出される。以上の文字成分表サーチの結果件数が０件であれば，第８図に示すようにここで０件という検索結果を出力して検索を終了する。また，‘湖’のように検索タームがただ１個の文字から構成される場合も，ここで文字成分表サーチの結果を出力して検索を終了する。もし，検索タームが複数の文字で構成されていて，かつ文字成分表サーチの結果件数が０件でなければ，次に凝縮本文サーチを行う。凝縮本文に登録されている内容は，文字種ごとに分割された文字列である。例えば，“知的インタフェース”のように，途中で文字種が異なれば凝縮本文では部分文字列へ分解され，「知的，インタフェース」のように分割点にセパレータが入る。したがって，“知的インタフェース”のように異なる文字種から構成される検索タームの場合，このままでは凝縮本文をサーチしても該当する文字列が存在しないことになる。そこで，凝縮本文サーチに入る前に検索タームをチェックし，異なる文字種で構成される検索タームはこれを文字種毎に分割する。このように文字種で分割するという処理を施した検索タームを元々の検索タームと区別して，分割検索タームと呼ぶ。そして凝縮本文サーチは，例えば“計算機”，“知的”，“インタフェース”のように分割検索タームで検索する。ただし，分割検索タームに関しては，分割元を同じくするターム間でＡＮＤ条件で検索を行う。例えば，「“計算機”ＯＲ“知的インタフェース”」という条件式の場合，凝縮本文サーチでは「“計算機”ＯＲ（“知的”ＡＮＤ“インタフェース”）」すなわち，「“知的”と“インタフェース”が同一文書内に存在するか，または“計算機”が存在する文書を探せ」という条件式として検索を行うことになる。凝縮本文サーチの結果が０件であれば，ここで０件という検索結果を出力して検索を終了する。　また近傍条件，または文脈条件の指定の有る場合，あるいは“知的インタフェース”の

ような分割される検索タームがある場合，つまり検索タームと分割検索タームが異なる場合に限り本文サーチを行う。そうでない場合，ここで階層型プリサーチを終了し凝縮本文の結果を検索結果として出力する。ここで，文脈条件とは例えば，「“計算機”［Ｓ］“知的インタフェース”」のように示される条件式でこれは，「“計算機”と“知的インタフェース”が同一の文（センテンス）内にあるものを探せ」という意味を表す。あるいは近傍条件とは，例えば，「“計算機”　［１０Ｃ］“知的インタフェース”」のように記述されるもので，これは，「“計算機”と“知的インタフェース”が１０文字以内に近接して現れる文書を探せ」という意味を表す。すなわち，文脈条件，近傍条件とも文書中に出現する検索タームの位置関係を指定する検索条件のことである。このような本文中に現れる検索タームの位置関係を指定した検索条件が与えられた場合，もしくは凝縮本文中ではセパレータで区切られた途中で文字種の変わる検索タームが与えられた場合には，凝縮本文サーチの結果に対応する本文データを参照し，与えられた条件通りに本文中に検索タームが出現するものを検索結果として出力し，検索を終了することになる。このように，検索タームが異なる文字種で構成されている場合，或いは検索ターム間の本文中での出現位置に関する条件指定がある場合についても，効率的に階層型プリサーチを行い，高速なフルテキストサーチを実現することができる。

【００１７】次に，本発明の第四の実施例について説明する。本実施例は，第一の実施例における文字成分表の容量を削減し，コンパクトにしたものである。第一の実施例で用いた文字コード依存型文字成分表は，処理が簡単であるが，文字成分表の１文書あたりのビットリストが長いため文字成分表が大きくなるという問題がある。また，該当する文字コードが存在しないのにエントリ番号を割当てているためむだな部分が多いという問題がある。例えばシフトＪＩＳの場合，（０００００）Ｈから（８１４０）Ｈの間，及び（Ａ０００）Ｈから（Ｅ０４０）Ｈの間，つまり０番目から３３０８７番目までと４０９６０番目から５７４０８番目までのエントリ番号には該当する文字コードがない。それにもかかわらず，文字コードによってエントリ番号を決定するためにこの部分も全て表のエントリとして持っている必要がある。このビットリスト中のむだな部分を排除するために一旦文字コードを変換し，ビット位置を０番目からすきまなく使用できるように文字成分表を作成する。この文字コード変換型文字成分表を用いた実施例の詳細について以下説明する。

【００１８】文字コード変換型文字成分表を作成するための文字コードへの変換式の例として次式をあげる。また，対応するＰＡＤ図を図１０に示す。

```
if SJIS < (A000)H then SCODE = SJIS − (8040)H
else SCODE = SJIS − (C040)H
SCODE = SCODE − (SCODE／256)×64
```

・・・・・・・　（４－１）

式（但し、通常文字コードの小さい値の部分は制御コードとして用いることが多いために、本式では（８１４０）Ｈとはせずに（８０４０）Ｈとして多少の余裕を持たせている。また、（ＳＣＯＤＥ／２５６）の演算結果の小数点以下は切り捨て、切り捨てた結果と６４との乗算を行う。）式中でＳＪＩＳがもとのシフトＪＩＳコードを示し，ＳＣＯＤＥは変換後の文字コードを示す。ＫＥＩＳコードや他のコード体系についてもシフトＪＩＳコードとの対応がとれているので同様の式でＳＣＯＤＥへの変換が可能である。（４－１）式は，文字コード表に表すと図１１のような変換を意味している。すなわち，（０００００）Ｈから（ＦＦＦＦ）Ｈまでの間に（８１４０）Ｈ～（９ＦＦＣ）Ｈ　及び　（Ｅ０４０）Ｈ～（ＦＥＦＣ）Ｈ　と分散して配置されている文字コードを，（０００００）Ｈ　からすきまなく配置するように文字コードを変換することになる。この（４－１）式を用いてコード変換することにより，図１２に示すようにビットリストの長さを非常に短くすることができ，文字成分表の全体の容量を小さくすることができる。

【００１９】階層型プリサーチの制御は，第一の実施例と同じである。すなわち，図８の制御手順をそのまま使用し，第１に検索ターム中の文字を使い文字成分表サーチを行い，第２に検索タームを用いて凝縮本文サーチを行う。文脈条件の指定がなければここで検索結果を出力し，検索を終了する。文脈条件の指定があれば第３に本文サーチを行いその結果を出力する。但し，文字成分表サーチのときには入力された検索タームは全て（４－１）式に基づいて文字コード変換を施して用いることになる。以上，文字コード変換型文字成分表を用いた第四の実施例について説明した。本実施例によれば，文字コードをコード変換し，ビット位置を０番目からすきまなく並べた文字成分表を作成することにより，文字成分表の文字の割り振られていないエントリを無くすことができ，文字成分表の容量を非常に小さくすることができる。

【００２０】次に，本発明の第五の実施例について説明する。本実施例は，第四の実施例における文字成分表の容量をハッシング手法を用いてさらに削減したものである。第四の実施例の文字成分表の容量をさらに小さくするために，本実施例ではビットリスト中の一つのエントリ番号に複数の文字を割り当てる。すなわち，ハッシュ関数を用いて検索ターム中の文字とビットリスト中のビット位置を対応付ける方法をとる。このハッシュ関数として例えば次の式を用いることができる。

```
h (SCODE) = mod (SCODE, N)
```

・・・・・　（５－１）式式中でＳＣＯＤＥは（４－

１）式によってシフトＪＩＳから変換した文字コードである。ｍｏｄは第１引き数を第２引き数で割った余りを出力する関数である。Ｎは任意の整数値である。Ｎとして，例えば５１２を用いると，‘あ’はエントリ番号４８０，‘ま’はエントリ番号１９３となる。

【００２１】このようにして作成した文字成分表の例を図１３に示す。この場合は，Ｎを５１２と設定したが，１文書を登録するのに５１２ビットしか必要としないことが分かる。検索時には，与えられた検索タームの各文字について登録時と同じように，（５－１）式のハッシュ関数を用いてエントリ番号を求め，これに対応する文字成分表のビット位置を参照する。例えば，“あいまい”という文字列の場合図１３のようにエントリ番号４８０，４８２，１９３の位置のビットがすべて１の文書を文字成分表サーチの検索結果とする。こうして文字成分表サーチで求められた文書について，次にその凝縮本文をサーチする。

【００２２】以下，凝縮本文サーチ及び本文サーチの制御手順について，図１４を用いて説明する。第一の実施例では，文字成分表サーチの後検索タームが一文字からなる場合には，文字成分表サーチの結果を検索結果として出力して階層型プリサーチを終了していた。しかし，この本実施例で用いた文字成分表の文字成分表サーチでは，検索ノイズの生じる可能性があるために，凝縮本文サーチまで階層型プリサーチを継続する必要がある。例えば，ひらがなの‘は’（シフトＪＩＳコード（８２ＣＤ）Ｈ）は，（５－１）式でエントリ番号１３であるが，漢字の‘艦’（シフトＪＩＳコード（８ＡＣＤ）Ｈ）も同じエントリ番号１３となる。このことは，検索タームとして“艦”が与えられた場合，“は”を含む文書もすべて文字成分表サーチの結果として検索されてくることになる。したがってさらに，凝縮本文をスキャンして実際に漢字の“艦”を含む文書を抽出し，これを検索結果として出力することになる。以上，第五の実施例について説明した。本実施例ではハッシュ関数を使って，文字成分表の１エントリに複数個の文字を割り当てることにより，文字成分表の容量を格段に小さくできるという効果が得られる。

【００２３】次に第六の実施例について説明する。第五の実施例のように単純にハッシングした場合，ひらがなのように文書中に出現しやすい文字と，ＪＩＳ第２水準の漢字のようにめったに出現しない文字とが同じエントリ番号となる可能性がでてくる。例えば，ひらがなの‘は’と，漢字の‘艦’は同じエントリ番号１３となり，検索タームとして“艦”が与えられた場合‘は’を含む文書はすべて文字成分表サーチの結果としてヒットすることになる。ひらがなの‘は’は日本語の文書では非常に使用頻度の高い文字のためほぼ全件の文書が文字成分表サーチでヒットする。このように文字成分表サーチでの絞り込みの率が低下すると，凝縮本文もスキャンする文書量が増えるために全体の検索処理時間が増大することになる。

【００２４】このような絞り込み率の低下を防ぐためには，ハッシュ関数を文字の使用頻度を考慮して定める必要がある。本実施例で用いる文字成分表を文字種別ハッシング型文字成分表と呼ぶ。文字種別ハッシング型文字成分表を作成するには，例えば図１５に示すように，各文字種毎に文字成分表のエントリ領域を割り当て，その領域内で文字コードにより折り返すようなハッシュ関数を作る。このようなハッシュ関数を実現するには，文字コードによって文字種を判定した後，ｍｏｄ関数で折り返してもよいし，文字コードとエントリ番号との対応表により実現することもできる。このハッシュ関数の一例を図１６にＰＡＤ図で示す。本実施例では，ひらがな，カタカナ，英字のエントリ数をそれぞれ２０とし，記号のエントリ数を１０，数字のエントリ数を１０，ＪＩＳ第１水準のエントリ数を３７０，ＪＩＳ第２水準のエントリ数を６１としている。まず，入力された検索タームに対して，文字コードにより文字種を判定し，それぞれの文字種ごとに文字成分表の割り当てられたエントリの部分をｍｏｄ関数を用いて折り返す。

【００２５】すなわち，ＳＣＯＤＥが（０１ＤＦ）Ｈから（０２３１）Ｈの範囲にあれば，ひらがな文字列であるので，そのＳＣＯＤＥを２０でｍｏｄをとってこれをエントリ番号とする。ＳＣＯＤＥが（０２４０）Ｈから（０２９６）Ｈの範囲にあれば，カタカナ文字列であるので，そのＳＣＯＤＥを２０でｍｏｄをとって，これにカタカナのハッシング領域の先頭である２０を足した値をエントリ番号とする。ＳＣＯＤＥが（０１Ａ０）Ｈから（０１ＤＡ）Ｈの範囲にあれば，英字文字列であるので，そのＳＣＯＤＥを２０でｍｏｄをとって，これに英字のハッシング領域の先頭である４０を足した値をエントリ番号とする。ＳＣＯＤＥが（０１８Ｆ）Ｈから（０１９８）Ｈの範囲にあれば，数字文字列であるので，そのＳＣＯＤＥを１０でｍｏｄをとって，これに数字のハッシング領域の先頭である７０を足した値をエントリ番号とする。ＳＣＯＤＥが（０６５Ｆ）Ｈから（１２３２）Ｈの範囲にあれば，ＪＩＳ第１水準の漢字文字列であるので，そのＳＣＯＤＥを３７０でｍｏｄをとって，これにＪＩＳ第１水準の漢字文字列のハッシング領域の先頭である８０を足した値をエントリ番号とする。ＳＣＯＤＥが（１２５Ｆ）Ｈから（１ＦＤＥ）Ｈの範囲にあれば，ＪＩＳ第２水準の漢字文字列であるので，そのＳＣＯＤＥを６１でｍｏｄをとって，これにＪＩＳ第２水準の漢字文字列のハッシング領域の先頭である４５０を足した値をエントリ番号とする。以上のＳＣＯＤＥ以外の場合には，記号その他の文字種による文字列とみなし，そのＳＣＯＤＥを１０でｍｏｄをとって，これに記号のハッシング領域の先頭である６０を足した値をエントリ番号とする。

【００２６】この文字種別ハッシング型文字成分表を用いた階層型プリサーチの制御手順は，第五の実施例と同じである。すなわち，第１に検索ターム中の文字を用いて文字成分表サーチを行い，第２に検索タームを用いて凝縮本文サーチを行う。文脈条件等が指定されていない場合には，ここで検索を終了するが，そうでない場合には，第３に本文サーチを行い結果を出力する。以上説明したように，本実施例によれば，使用頻度を考慮して文字種ごとに文字成分表のエントリ番号を対応させた文字種別ハッシング型文字成分表を用いることにより，文字成分表サーチでのノイズを少なくできるため，凝縮本文における文書のスキャン量が減り，その分高速なフルテキストサーチが可能となる。

【００２７】次に第七の実施例として，さらに文字成分表サーチにおける絞り込みの率を向上させ，凝縮本文のスキャン量を減らすことのできる頻度情報ハッシング型文字成分表を用いた階層型プリサーチの制御方法を説明する。頻度情報ハッシング型文字成分表を作成するには，データベースに登録してある文書の文字の使用頻度を調べ，頻度情報によりハッシュ関数を決定する。頻度の大きい文字については，同一エントリにできるだけ他の文字が入らないようにし，頻度の少ない文字について同一エントリに複数個の文字が入るようにハッシュ関数を調整する。こうすることにより，平均的に安定した絞り込み率が文字成分表サーチで得られることになる。具体的には，図１７に示すように（４－１）式で得られるＳＣＯＤＥをもとに一度データベース中で該当する文字を使用している文書数を調べ頻度順に並べ替える。次に，頻度の大きいものから文字成分表のエントリ数分Ｎｔだけとる。そしてＮｔ以内の頻度数分布のうち最も上位の頻度を持つエントリだけを残して，その他のエントリに順次Ｎｔ以上のエントリ番号を割り付けていく。このＮｔ以上のエントリ番号の割付けには（Ｎｔ＋１）番目のエントリをＮｔのエントリとし，（Ｎｔ＋２）番目を（Ｎｔ－１）番目のエントリとするように，Ｎｔより順次頻度の大きいエントリを割り付けていく。割り付けていく過程では，常に最上位の頻度を持つエントリの上には，他のエントリを割り付けないようにする。割り付けたエントリは，図１８に示すようにテーブルの形で，記憶しておきこのテーブルを参照してハッシュ関数を構成する。すなわち，ＳＣＯＤＥが（０９５Ｆ）Ｈの文字‘検’は，エントリ番号２３１であることが分かる。

【００２８】階層型プリサーチの制御手順は，第五の実施例と同じである。すなわち，図１４の制御手順をそのまま使用し，第１に検索ターム中の文字を用いて文字成分表サーチを行い，第２に検索タームを用いて凝縮本文サーチを行う。文脈条件等が指定されていない場合には，ここで検索を終了するが，そうでない場合には，第３に本文サーチを行い結果を出力する。このように，本実施例によれば；データベース中で実際に用いられる文字の頻度分布をもとに文字成分表を作成することによって，文字成分表サーチで常に安定して高い絞り込み率が得られるため，検索タームに依存せず安定して短時間の検索処理時間を得ることができる。

【００２９】以上，文字成分表の異なる実施例について五つの実施例を説明した。これより凝縮本文の異なる実施例についての説明をする。第一の実施例で用いた凝縮本文は作成の処理が簡単であるが，図４でも分かるように“のための”というような本来検索に使われないような文字列まで凝縮本文に残ることになる。このことは凝縮本文の圧縮率低下を招く。つまり，検索時にスキャンする凝縮本文の量が増えるため，検索処理時間が増加してしまう。このような，凝縮本文の圧縮率を低下させる主な要因は，“のための”というような付属語の連なったそれ自体では意味を持たない文字列を凝縮本文に登録してしまうところにある。

【００３０】そこで，第八の実施例として，この検索に不要な付属語の連なりを除去した凝縮本文を用いる階層型プリサーチを説明する。この凝縮本文を文字種分割・重複排除・付属語除去型凝縮本文と呼ぶ。この凝縮本文の作成方法は図１９に示すように，本文のテキスト文字列から文字種分割して部分文字列に分け，それから重複語を排除した後，付属語の除去を行う。文字種分割と重複排除は第一の実施例と変わらない。付属語除去は，重複排除の済んだひらがな文字列に対して行う。この付属語除去のための解析は，図２０に示すように基本単語辞書と単語間の接続規則を基に行う。基本単語辞書には，図２１のようにひらがなのみから構成される動詞，指示代名詞，形容詞，形容動詞，副詞，接続詞，助詞，助動詞，またこれらの品詞の活用語尾が品詞情報とともに登録されている。本図の例では，動詞として＜ある＞，＜なる＞，＜もつ＞等がそれらの活用語尾とともに登録されている。接続規則には基本単語辞書に登録された各語が他のどの語と接続し得るかを登録する。例えば図２２に示すように，＜動詞－もつ連体形＞に＜名詞－こと＞が接続し，さらに＜名詞－こと＞には＜助詞－が＞が接続し得ることが登録されている。このような基本単語辞書及び接続規則を用いてひらがなの部分文字列が付属語から構成されているか否かを判定し，凝縮本文へその文字列を登録するか否かを決定する。例えば，“のための”という部分文字列は＜助詞－の＞＜名詞－ため＞＜助詞－の＞というように接続した文字列と解析できるため，付属語のみから構成された文字列と判定し排除する。一方，“あいまい”という文字列は，付属語と解析ができないため排除せずにそのまま凝縮本文へ登録する。

【００３１】このように，付属語を解析してひらがな文字列を排除し，検索に使われることのない無用の情報を削除することによって，凝縮本文の圧縮率を高めることが可能となる。また解析に用いる基本単語辞書と接続規

則は，時代とともに登録語数が増えていく従来のキーワード辞書とは基本的に異り，普遍的なもので一度作成してしまえば更新していく必要がないという利点がある。付属語として解析できるものだけを排除するために，辞書に存在しないひらがなから構成される新語が現れても必ず凝縮本文に残るということになる。

【００３２】次に，文字種分割・重複排除・付属語除去型凝縮本文を用いた階層型プリサーチ方式の制御について説明する。文字種分割・重複排除・付属語除去型凝縮本文では，ひらがな文字列を付属語解析して凝縮本文に登録しない場合がある。そのため，特定のひらがな文字列で検索しようとした場合，凝縮本文サーチで検索もれとなる場合がある。例えば“めまい”という文字列は，動詞の未然形活用語尾“め”と助動詞“まい”の終止形が接続したものと解析できる。具体例としては，“認めまい”があげられる。ところが“めまい”は，名詞として使われている場合でも，付属語除去処理の結果凝縮本文からは削除されてしまう。したがってこのような場合，“めまい”で凝縮本文を検索すると検索もれが生じる可能性がでてくる。そのため，検索タームが凝縮本文中にもともとない言葉なのか，あるいは凝縮本文作成過程で除去された可能性のある言葉なのかをチェックしてから検索する必要が生じる。検索タームが凝縮本文に登録されるべき語か否かというチェックは，凝縮本文を作成したときに用いた付属語除去のアルゴリズムをそのまま適用する。この例では，“めまい”という検索タームが与えられたときは，これが付属語の連なりと判定することができる。

【００３３】以上の検索制御の手順を図２３で説明する。まず文字成分表サーチを行う。結果件数が０件であれば，０件を検索結果として出力して検索処理を終了する。第一の実施例でも述べたが，ハッシュ関数を用いない方式では検索タームが一文字の場合にかぎり，文字成分表のサーチ結果を最終検索結果として出力できる。すなわち，第一及び第四の実施例で説明した文字成分表を用いる場合には，検索タームが一文字であるか否かを調べ，一文字であれば文字成分表サーチの結果を検索結果として出力し，処理を終了する。第五，第六，第七の実施例で述べたハッシュ関数による文字成分表を用いる場合には，この検索タームが一文字か否かというチェックは行わず，常に次の凝縮本文サーチを行う。この後，第一の実施例と同様に，分割検索タームを生成する。

【００３４】次に，分割検索タームのそれぞれについて付属語解析を行う。分割検索タームのうち一つでも付属語と判定された場合，その分割検索タームは凝縮本文から削除されている可能性があるので，凝縮本文サーチを行わず，文字成分表サーチの結果に基づいて本文を直接サーチする。一方，付属語解析の結果，分割検索タームが全て付属語でないと判定されたならば，第一の実施例と同様に凝縮本文サーチを行う。近傍条件あるいは，文脈条件の指定がない場合，あるいは分割検索タームがもとの検索タームと同じ場合には，この凝縮本文サーチの結果を最終検索結果として出力し，検索を終了する。もし，近傍条件ないし文脈条件が指定されている場合，あるいは分割検索タームと元の検索タームが異なる場合には，次に本文サーチを実行し，その結果を最終的な検索結果出力とする。このように，本実施例によれば，ひらがな文字列を解析し，不要な付属語の連なりを凝縮本文から除去した文字種分割・重複排除・付属語除去型凝縮本文を用いることにより，凝縮本文の圧縮率を向上させ，検索処理時間を短縮することができる。

【００３５】次に，第九の実施例として，ひらがな文字列を全て排除した，文字種分割・重複排除・ひらがな文字列除去型凝縮本文を用いる階層型プリサーチを説明する。第八の実施例で説明した凝縮本文は，確かに圧縮率が上がるものの付属語解析の際に誤った解析をする可能性がある。例えば第八の実施例でも用いた“めまい”という文字列の例の外にも，付属語解析だけでは本質的にどれが付属語か正しく判定できない場合がまれにある。例えば，“動作してこの応用で．．．”という文書の中の“してこの”という部分文字列は，，“～して，この～”という意味で用いられているのか，“～し，てこの～”のように機械のてこを意味しているのかが判定するのが難しい。後者の意味で用いられている場合には，“てこ”という検索タームを指定した際に，“てこ”は付属語と判定されないため，凝縮本文をサーチしにいくことになる。一方，凝縮本文作成では，“してこの”が付属語と解析され凝縮本文から削除されているため凝縮本文サーチで検索もれとなってしまう。この付属語解析の不完全さを補正するために，ひらがな文字列か否かという単純な判定方法で階層型プリサーチを実現するのが，本第九の実施例である。この凝縮本文の作成方法を，図２４に示す。本方法では文字種分割の後，ひらがなを除去して重複登録排除を行う。

【００３６】この文字種分割・重複排除・ひらがな文字列除去型凝縮本文を用いた階層型プリサーチの制御手順について図２５を用いて説明する。まず第八の実施例と同様に文字成分表サーチを行う。この後，分割検索タームを生成する。次に，分割検索タームのそれぞれについてひらがな文字列か否かチェックを行う。分割検索タームのうち一つでもひらがな文字列がある場合，凝縮本文サーチを行わず，文字成分表サーチの結果に基づいて本文を直接サーチする。一方，分割検索ターム中にひらがな文字列がない場合，第一の実施例と同様に凝縮本文サーチを行い，近傍，文脈条件の指定がある場合，あるいは分割検索タームが元の検索タームと異なる場合には，本文サーチまで検索処理を続行する。このように，本実施例によれば，ひらがな文字列を全て排除した凝縮本文を用いることによって，ひらがな文字列についても検索もれのない正確なフルテキストサーチが実現できる。

【００３７】次に，本発明の第十の実施例について，説明する。上記第九の実施例では，ひらがなの検索タームが与えられた場合，本文を直接参照する必要がある。したがって検索時間がより多く掛かることになる。そこで，ひらがなの検索タームが与えられた場合でも高速にフルテキストサーチできる方法として，第十の実施例の説明をする。本実施例では，第九の実施例で用いた凝縮本文の外に第九の実施例では除去したひらがな文字列を登録した凝縮本文を別に作成する。図２６に示すように，文字種分割，重複登録排除の後，残った部分文字列がひらがな文字列か否かを判定し，ひらがな文字列以外を凝縮本文Ａとして登録し，ひらがな文字列を凝縮本文Ｂとして登録する。こうすれば，ひらがなだけの検索タームが与えられた際，凝縮本文Ｂを探索することができるようになるため，検索時間を短縮することが可能となる。実際の階層型プリサーチの検索制御の手順を図２７に示す。まず第八の実施例と同様に文字成分表サーチを行う。もし，検索結果が０件なら，ここで検索を終了する。この後，分割検索タームを生成する。次に，分割検索タームをひらがな文字列のタームとそれ以外の文字列からなるタームに分類する。その後，ひらがな以外の文字列からなる分割検索タームがある場合には，凝縮Ａをサーチする。次にひらがなの分割検索タームがある場合には，凝縮Ｂをサーチする。その後は，第一の実施例と同様に，近傍，文脈条件の指定がある場合，あるいは分割検索タームがもとの検索タームと異なる場合には，本文サーチまで検索処理を続行する。このように，ひらがなのみの凝縮本文と，ひらがな以外の凝縮本文と分けて格納することにより，どんな文字種の検索タームが入力されても，凝縮本文を有効に活用でき，常に高速なフルテキストサーチが実現できる。

【００３８】次に，第十一の実施例について説明する。本実施例は，凝縮本文の圧縮率を上げるために，文字種毎に独立した凝縮本文を用いる方法に基づいたものである。本実施例で用いる凝縮本文を文字種分割・重複排除・文字種別登録型凝縮本文と呼ぶ。この文字種分割・重複排除・文字種別登録型凝縮本文を作成するには，図２８に示すように，文字種分割，重複登録排除を行った後，残った部分文字列の文字種を判定してひらがな凝縮本文Ｈ，カタカナ凝縮本文Ｉ，漢字凝縮本文Ｊ，英字凝縮本文Ｋ，数字凝縮本文Ｌ，記号その他の文字種凝縮本文Ｍに分類して登録する。こうすることにより，例えば漢字の検索タームで検索する場合には，漢字文字種の凝縮本文Ｊのみをサーチすればよいことになるため，検索時間をさらに短縮することができる。具体的な階層型プリサーチの制御手順を図２９を用いて説明する。まず，第八の実施例と同様に文字成分表サーチを行う。検索結果件数が０件なら，ここで検索を終了する。この後，分割検索タームを生成する。次に，分割検索タームを文字種毎に分類する。その後，ひらがなの分割検索タームがある場合には凝縮Ｈを，カタカナの分割検索タームがある場合には凝縮Ｉを，というように分解検索タームの文字種にしたがってサーチする凝縮本文を選択する。その後は，第一の実施例と同様に，近傍，文脈条件の指定がある場合，あるいは分割検索タームがもとの検索タームと異なる場合には，本文サーチまで検索処理を続行する。このように，文字種ごとに凝縮本文ファイルを分離し個々の凝縮本文の容量を小さくすることにより，漢字のみ，カタカナのみ，あるいはひらがなのみ，といった単一文字種の検索タームでのフルテキストサーチが高速に行えるという効果が得られる。

【００３９】次に第十二の実施例について，図３０および図３１を用いて説明する。本実施例は，特願平０２－１９３０１５で提案した文書検索装置を用い，本発明を実現したものである。本装置の主な構成は，キーボード３００１，検索式解析プログラム３００２，ビットサーチプロセッサ３００７ａ，ストリングサーチエンジン３００６，複合条件判定用マイクロプロセッサ３０４５ａ，検索結果格納メモリ３０４６，ディスプレイ３０２０，半導体メモリ装置３０１０ａ，ＲＡＭディスク装置３０１０ｂ，集合型磁気ディスク３０１０ｃ，及び検索実行制御プログラム３００８よりなる。半導体メモリ装置３０１０ａには文字成分表が，ＲＡＭディスク装置３０１０ｂには凝縮本文，集合型磁気ディスク装置３０１０ｃには本文がそれぞれ格納されている。但し，文字成分表及び凝縮本文は，集合型磁気ディスク３０１０ｃに格納されていて，本装置の運用開始時点でそれぞれ半導体メモリ装置３０１０ａ及びＲＡＭディスク装置３０１０ｂへローディングされる。

【００４０】階層プリサーチ制御の手順は，いままで実施例で説明してきたものと変わらない。いままでの実施例との相違点は，文字成分表を半導体メモリ，凝縮本文をＲＡＭディスク，本文を集合型磁気ディスクに格納したところと，文字成分表サーチ専用のマイクロプロセッサ，凝縮本文サーチ及び本文サーチ専用のストリングサーチエンジンを用いていることである。検索処理の手順を以下に説明する。

【００４１】キーボード３００１から入力した検索条件式はサーチマシン制御用マイクロプロセッサＭＰＵ０３０５０上の検索式解析プログラム３００２により解析される。すなわち、検索式解析プログラム３００２では検索条件式を構成するキーワード部分とそれらの包含条件及び配置条件を記述した複号条件記述部に分離する。包含条件は論理条件として記述され、配置条件は近傍条件や文脈条件として記述されたものである。分離抽出後、キーワード部分は同じくＭＰＵ０３０５０上の同義語展開プログラム３００３に渡され、複号条件記述部は複号条件解析プログラム３０４１に渡される。同義語展開プログラム３００３では、ここに内蔵された同義語辞書を参照して、入力されたキーワードの同義語が求められ

る。そして、ここで同義語展開されたキーワード群は異表記展開プログラム３００４へ渡される。本図の例の場合、“計算機”から、“電算機”、“コンピュータ”、“ＣＯＭＰＵＴＥＲ”などが生成される。異表記展開プログラム３００４では、ここに入力されてきたキーワード群に対して異表記展開処理が施される。本図の例の場合、“コンピュータ”から“コンピューター”が、“ＣＯＭＰＵＴＥＲ”から“Ｃｏｍｐｕｔｅｒ”などが生成される。こうして同義語及び異表記展開されたキーワード群は、次にオートマトン生成用マイクロプロセッサＭＰＵ１３００５ａ上のオートマトン生成用プログラム３００５に送られる。オートマトン生成用プログラム３００５では、異表記展開プログラム3.００４から送られてきたキーワード群に対して、これらを一括照合するオートマトンを生成し、状態遷移テーブルと照合すべきキーワードの識別コード情報として、サーチエンジン３００６に設定する。サーチエンジン３００６は有限オートマトン方式に基づく高速多重文字照合回路である。また、異表記展開プログラム３００４で異表記展開されたキーワード群は、該当キーワードと共に、ビットサーチ用マイクロプロセッサＭＰＵ３３００７ａ上のビットサーチプログラム３００７へ渡される。

【００４２】一方，近傍条件，文脈条件や，ＡＮＤ，ＯＲ等の論理条件は検索式解析プログラム３００２から，複合条件解析プログラム３０４１，近傍条件解析プログラム３０４２，文脈条件解析プログラム３０４３，論理条件解析プログラム３０４４を経て複合条件判定プログラム３０４５へと送られる。必要な検索情報がビットサーチプログラム３００７，ストリングサーチエンジン３００６，複合条件判定プログラム３０４５へ送られた後，検索制御実行プログラム３００８は，まずビットサーチプログラム３００７に起動を掛ける。ビットサーチプログラム３００７は，半導体メモリ装置３０１０ａに格納してある文字成分表を読み出し，文字成分表サーチを行う。文字成分表サーチの結果は，検索結果格納メモリ３０４６へ格納する。

【００４３】文字成分表サーチが終った後，検索実行制御プログラム３００８は，検索結果格納メモリ３０４６を参照し，検索結果が０件であれば，０件を検索結果として出力し検索処理を中断する。検索結果が０件でなければ，ストリングサーチエンジン３００６へ起動をかけると同時に検索結果格納メモリ３０４６に格納されている文字成分表サーチの結果でヒットした文書の凝縮本文をＲＡＭディスク装置２９１０ｂから読み出し，ストリングサーチエンジン３００６へ送り，凝縮本文サーチを実行させる。この結果件数が０件であるか否かの条件判定は検索実行制御プログラム３００８で行う。ストリングサーチエンジン３００６では，ＲＡＭディスク装置３０１０ｂより読み出された，凝縮本文を分割検索タームでサーチする。照合結果は複合条件判定プログラム３０４５に順次送られる。複合条件判定プログラム３０４５では，検索ターム間に付与された論理条件を判定し，条件に適合する文書の文書番号を検索結果格納メモリ３０４６へ順次格納する。

【００４４】凝縮本文サーチが終了した後，検索実行制御プログラム３００８は，もう一度検索結果格納メモリ３０４６を参照し，結果件数が０件であれば，０件を検索結果として出力し，検索を終了する。０件でない場合で，近傍，文脈条件が設定されているか，もしくは分割検索タームが検索タームと異なっている場合にかぎり検索結果格納メモリから，検索結果文書番号を読み取り，これに対応する本文を集合型磁気ディスク装置３０１０ｃから読み出し，ストリングサーチエンジン３００６へ送り，今度は本文サーチを実行させる。近傍，文脈条件が設定されてなく，かつ分割検索タームが検索タームと等しい場合には，検索結果格納メモリに格納されている検索結果件数を出力し，検索を終了する。

【００４５】ストリングサーチエンジン３００６では，集合型磁気ディスク装置３０１０ｃから読み出された本文をスキャンして本文サーチを行う。結果は複合条件判定プログラム３０４５に順次送られる。複合条件判定プログラム３０４５では，検索ターム間に付与された論理条件のほか近傍，文脈条件を判定し，条件に適合する文書の文書番号を順次検索結果格納メモリ３０４６へ格納する。本文サーチまで実行した場合は，本文サーチの終了後，検索実行制御プログラム３００８は，検索結果格納メモリ３０４６を参照し検索結果件数を出力して検索を終了する。このように，容量の大きな本文データを磁気ディスクに，容量の小さな文字成分表や凝縮本文を，半導体メモリやＲＡＭディスクに格納することにより，大規模なデータベースに対しても高速なフルテキストサーチを実現することが可能となる。

【００４６】次に凝縮本文を磁気ディスクに格納した第十三の実施例について説明する。凝縮本文を磁気ディスクに格納する場合，階層型プリサーチの制御の手順を最適化することによって，同一の構成を用いた通常の階層型プリサーチを実行するよりも高速に処理することができる。以下，この制御の手順について説明する。磁気ディスクは通常，機械的に動く磁気ヘッドを持っている。このため，ディスク上の情報を飛び飛びに読み出す（スキップアクセスと呼ぶ）よりも，まとまった情報を一括して読み出す（シーケンシャルアクセスと呼ぶ）方が速いという特徴がある。いま，スキップアクセスの読み出し速度をＶｓｋｉｐ　ＭＢ／ｓ，シーケンシャルアクセスの読み出し速度をＶｓｅｑ　ＭＢ／ｓとすると，データベース全件の文書数をＮａ件，文字成分表サーチの結果件数をＮｃ件とし，文書の容量が均一であるとした場合，　Ｎｃ　＞　（Ｖｓｋｉｐ／Ｖｓｅｑ）・Ｎａ　……（１２－１）式のとき，シーケンシャルアクセスにより凝縮本文を全件サーチした方が，文字成分表サーチ

の結果に基づいてスキップアクセスするよりも処理時間が短くなる。したがって，図３２に示すように文字成分表サーチの後，階層プリサーチ制御プログラムにおいて結果件数を判定し，（１２－１）式を満たすヒット件数に達した場合には，文字成分表サーチの結果を無視して，凝縮本文をデータベース全件分サーチする。以上の方法を用いると，磁気ディスクに凝縮本文を格納するために，大容量のＲＡＭディスクを使用しなくともすみ，比較的高速なフルテキストサーチを低価格の文書検索装置で実現できることになる。

$$NcQsr/Vsr+NsrQtx/Vtx > NcQtx/Vtx$$

……（１３－１）式

のとき，凝縮本文サーチをせずに，本文サーチを直接行ったほうが検索時間が短くなる。Ｎｓｒ は凝縮本文を実際にサーチするまでわからないが，あらかじめ定数を

$$Nsr=\alpha Na \qquad (0<\alpha<1)$$

……（１３－２）式として，

（１３－１）式を変形すると，

$$Nc < \alpha Na\,(Qtx/Vtx)/(Qtx/Vtx-Qsr/Vsr)$$

……（１３－３）式

のとき，本文サーチを直接行うことにする。αをしきい値として検索前にあらかじめ値を設定しておき，文字成分表サーチの後（１３－３）式により凝縮本文サーチを行うか否か決定する。この制御を行うことにより，近傍，文脈条件の指定の下で高速なフルテキストサーチを実現することができる。以上，第十二の実施例の廉価版のシステム構成でフルテキストサーチを実現する第十三，第十四の実施例について説明した。

【００４８】このほかにも，凝縮本文をまったく使用せず凝縮本文サーチのステップを省いて，文字成分表サーチから直接本文サーチを実行する制御方法によっても階層型プリサーチを実現することができる。この方法によれば，本文をスキャンする量が増えるため検索時間は多少掛かるが，高価なＲＡＭディスクを使用しなくとも済み，また凝縮本文を格納する磁気ディスク容量が不要となるため，さらに低価格の文書検索装置を実現できることになる。また，文字成分表を使用せずに直接ＲＡＭディスクあるいは磁気ディスク上の凝縮本文を全件サーチし，近傍，文脈条件などの検索ターム間の位置関係の検索条件指定があるときにのみ本文サーチする制御方法によっても階層型プリサーチを実現することができる。この方法によれば，凝縮本文の探索量が増えるため検索時間は多少掛かるが，文字成分表を格納する半導体メモリが不要となるため，その分低価格の文書検索装置を実現できることになる。

【００４９】あるいは，今までの実施例で用いていたビットリスト形式の文字成分表を図３３に示すように，文書中に現れる文字を書き列ねた形式，すなわち１文字を１ビットとして表すのではなく，そのまま文字コード自体として格納した文字成分表を使用することもできる。あるいはこの時に，第五の実施例，第六の実施例，及び第七の実施例で説明したハッシュ関数を用いて一つの文字エントリに複数個の文字を対応させ文字成分表の容量を削減することもできる。このように文字コードを格納した文字成分表を用いた文字成分表サーチは，凝縮本文や本文サーチと同様に，一文字ずつファイルからデータを読み出し該当する文字が存在するか否か判定することで実現できる。このように，本文中で用いられている文字のみを集めた文字成分表を用いることにより，データ構造を簡素化でき，かつビット演算をせずに凝縮本文，本文サーチと同じスキャン型のサーチを用いることができるため，検索処理方法が簡素化できるという効果が得られる。

【００４７】次に凝縮本文を磁気ディスクに格納した第十四の実施例について説明する。近傍，文脈条件が指定されている場合には，文字成分表サーチ結果が非常に少ない場合，凝縮本文サーチを行わずに，文字成分表サーチ結果をもとに本文を直接サーチするほうが検索時間が短くなる。今，凝縮本文のサーチ速度を Ｖｓｒ ＭＢ／ｓ，本文のサーチ速度を ＶｔｘＭＢ／ｓ とし，文字成分表の結果件数を Ｎｃ，凝縮本文の結果件数を Ｎｓｒ，凝縮本文の１件当たりのデータ容量を Ｑｓｒ，本文の１件当たりのデータ容量を Ｑｔｘ とすると，

設定して凝縮本文サーチを行うか否か決定することになる。たとえば，データベース全体の文書数を Ｎａ として

【００５０】さらに，文字成分表も磁気ディスクに格納した構成でも，階層型プリサーチを実現することができる。この磁気ディスクに文字成分表を格納した場合には，文字成分表サーチにおいて検索ターム中で用いられている文字のビットリストを磁気ディスクから順次読み出しながらビット演算処理を行っていく。もしくは，上記の文字コードをそのまま文字成分表とした場合には，文字成分表を順次読み出しながら該当する文字を全て含む文書を選びだす。この文字成分表を磁気ディスクに格納する方法によれば，半導体メモリを使わずに済むために，さらに低価格の文書検索装置を実現することが可能となる。

【００５１】

【発明の効果】本発明によれば，文字成分表及び凝縮本文を用いて，階層的に文字レベル及び単語レベルで入力された検索タームに関連しない文書をふるい落すことにより，無用の本文サーチを省くことができるため，高速な文書検索フルテキストサーチの実現技術となり，大規模な文書データベースでも実用的な応答速度で文書検索

を行うことが可能となる。

【図面の簡単な説明】

【図１】本発明の第一の実施例の構成を示す図である。

【図２】本発明の特徴となる階層型プリサーチのための登録処理を示す図である。

【図３】本発明の特徴となる階層型プリサーチの検索処理を示す図である。

【図４】凝縮本文を作成する一例を示した図である。

【図５】凝縮本文の格納形態を示す図である。

【図６】文字成分表の概要を示す図である。

【図７】文字成分表サーチの概要を示す図である。

【図８】階層型プリサーチの処理手順を示す図である。

【図９】第三の実施例における文字成分表サーチの処理を示す図である。

【図１０】第四の実施例で用いる文字成分表のコード変換の処理を示すＰＡＤ図である。

【図１１】第四の実施例で用いる文字成分表のコード変換の概要を示す図である。

【図１２】第四の実施例で用いる文字成分表の概要を示す図である。

【図１３】第五の実施例で用いる文字成分表の概要を示す図である。

【図１４】第五の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図１５】第六の実施例で用いる文字成分表の概要を示す図である。

【図１６】第六の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図１７】第七の実施例で用いる文字成分表の作成方法の概要を示す図である。

【図１８】第七の実施例で用いる文字成分表のためのハッシュ関数で用いる文字コードーエントリ番号の対応表の概要を示す図である。

【図１９】第八の実施例で用いる凝縮本文の作成する方法を示す図である。

【図２０】第八の実施例で用いる凝縮本文のためのひらがな文字列の処理方法を示す図である。

【図２１】第八の実施例で用いる付属語解析のための基本単語辞書を示す図である。

【図２２】第八の実施例で用いる付属語解析のための接続規則を示す図である。

【図２３】第八の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図２４】第九の実施例で用いる凝縮本文の作成する方法を示す図である。

【図２５】第九の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図２６】第十の実施例で用いる凝縮本文の作成する方法を示す図である。

【図２７】第十の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図２８】第十一の実施例で用いる凝縮本文の作成する方法を示す図である。

【図２９】第十一の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図３０】第十二の実施例の構成の部分を示す図である。

【図３１】第十二の実施例の構成の残りの部分を示す図である。

【図３２】第十二の実施例で用いる階層型プリサーチの処理手順を示す図である。

【図３３】文字として格納した文字成分表の概要を示す図である。

【符号の説明】

１０３：本文

１０４：凝縮本文

１０５：文字成分表

２０１：本文登録プログラム

２０２：凝縮本文作成登録プログラム

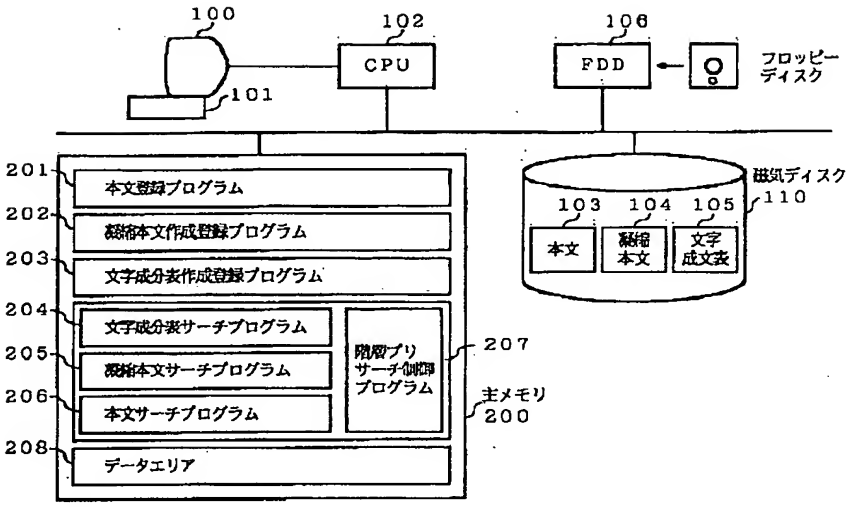
２０３：文字成分表作成登録プログラム

２０４：文字成分表作成検索プログラム

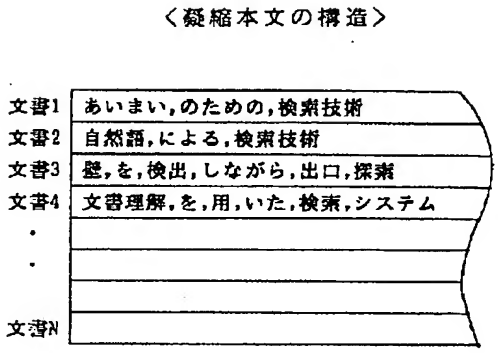
２０５：凝縮本文作成作成プログラム

２０６：本文サーチプログラム

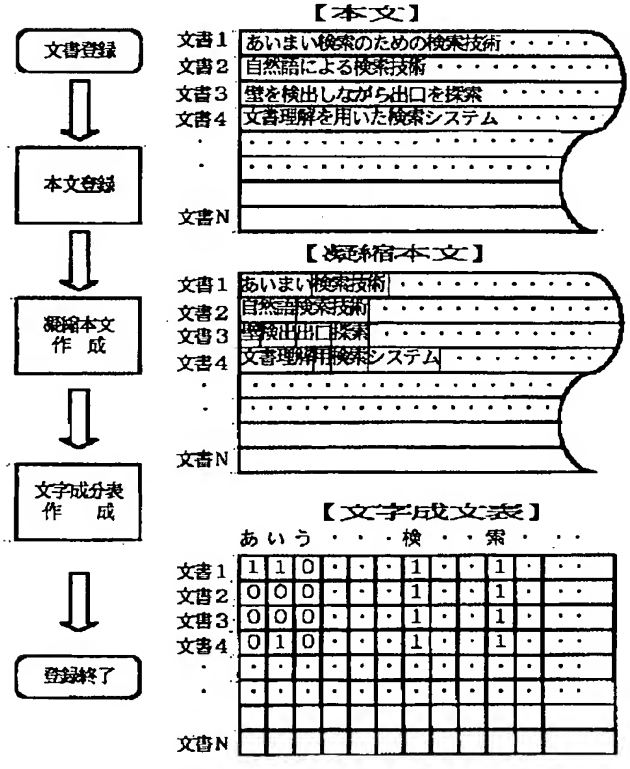
【図１】


【図１】
100
102
CPU
106
FDD
フロッピーディスク
101
201
本文登録プログラム
202
疑縮本文作成登録プログラム
203
文字成分表作成登録プログラム
204
文字成分表サーチプログラム
205
疑縮本文サーチプログラム
206
本文サーチプログラム
階層ブリサーチ制御プログラム
207
主メモリ
200
208
データエリア
磁気ディスク
110
103
104
105
本文
疑縮本文
文字成文表

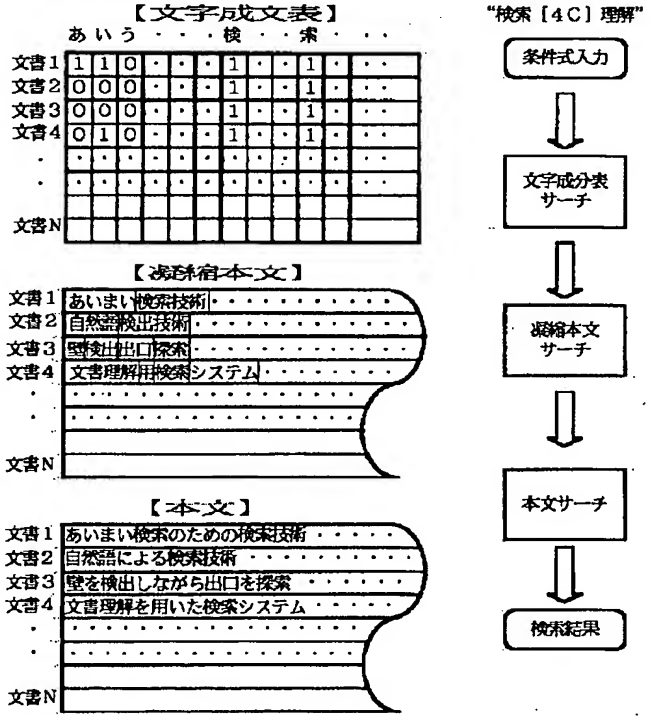
【図５】


【図５】
〈疑縮本文の構造〉
文書1
あいまい,のための,検索技術
文書2
自然語,による,検索技術
文書3
壁,を,検出,しながら,出口,探索
文書4
文書理解,を,用,いた,検索,システム
文書N

【図２】


【図2】
〈階層型ブリサーチにおける文書の登録処理方法〉
文書登録
本文登録
疑縮本文作成
文字成分表作成
登録終了
【本文】
文書1
あいまい検索のための検索技術
文書2
自然語による検索技術
文書3
壁を検出しながら出口を探索
文書4
文書理解を用いた検索システム
文書N
【疑縮本文】
文書1
あいまい検索技術
文書2
自然語検索技術
文書3
壁検出出口探索
文書4
文書理解用検索システム
文書N
【文字成文表】
あ い う ・・・ 検 ・・ 索 ・・・
文書1 1 1 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書2 0 0 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書3 0 0 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書4 0 1 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書N

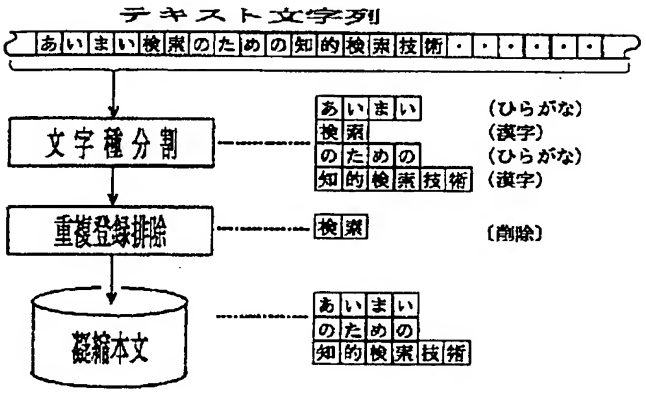
【図３】


【図3】
〈階層型ブリサーチにおける検索処理方法〉
【文字成文表】
あ い う ・・・ 検 ・・ 索 ・・・
文書1 1 1 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書2 0 0 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書3 0 0 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書4 0 1 0 ・ ・ ・ 1 ・ ・ 1 ・ ・
文書N
【疑縮本文】
文書1
あいまい検索技術
文書2
自然語検出技術
文書3
壁検出出口探索
文書4
文書理解用検索システム
文書N
【本文】
文書1
あいまい検索のための検索技術
文書2
自然語による検索技術
文書3
壁を検出しながら出口を探索
文書4
文書理解を用いた検索システム
文書N
"検索【4C】理解"
条件式入力
文字成分表サーチ
疑縮本文サーチ
本文サーチ
検索結果

【図４】

【図４】

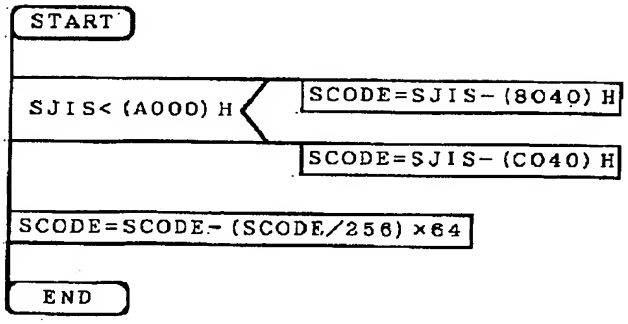
〈文字種別型・重複排除型縮約本文の作成方法〉



【図１０】

【図１０】

<文字コード変換アルゴリズム>



【図２１】

【図２１】

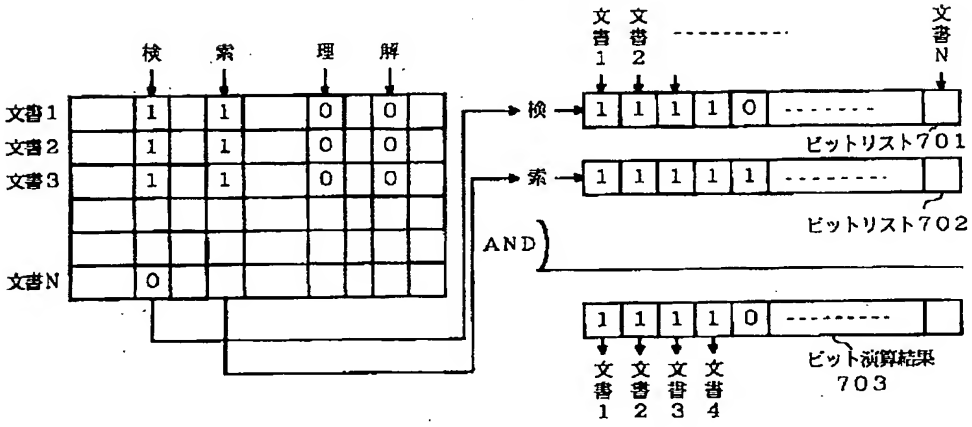
基本単語辞書

<動詞－ある連体形>	：ある
<動詞－ある仮定形>	：あれ
<動詞－なる未然形>	：なら
<動詞－なる連用形>	：なり
<動詞－もつ未然形>	：もた
<動詞－もつ連体形>	：もつ
<助詞－が>	：が
<名詞－こと>	：こと
<名詞－ため>	：ため
<名詞－の>	：の

【図７】

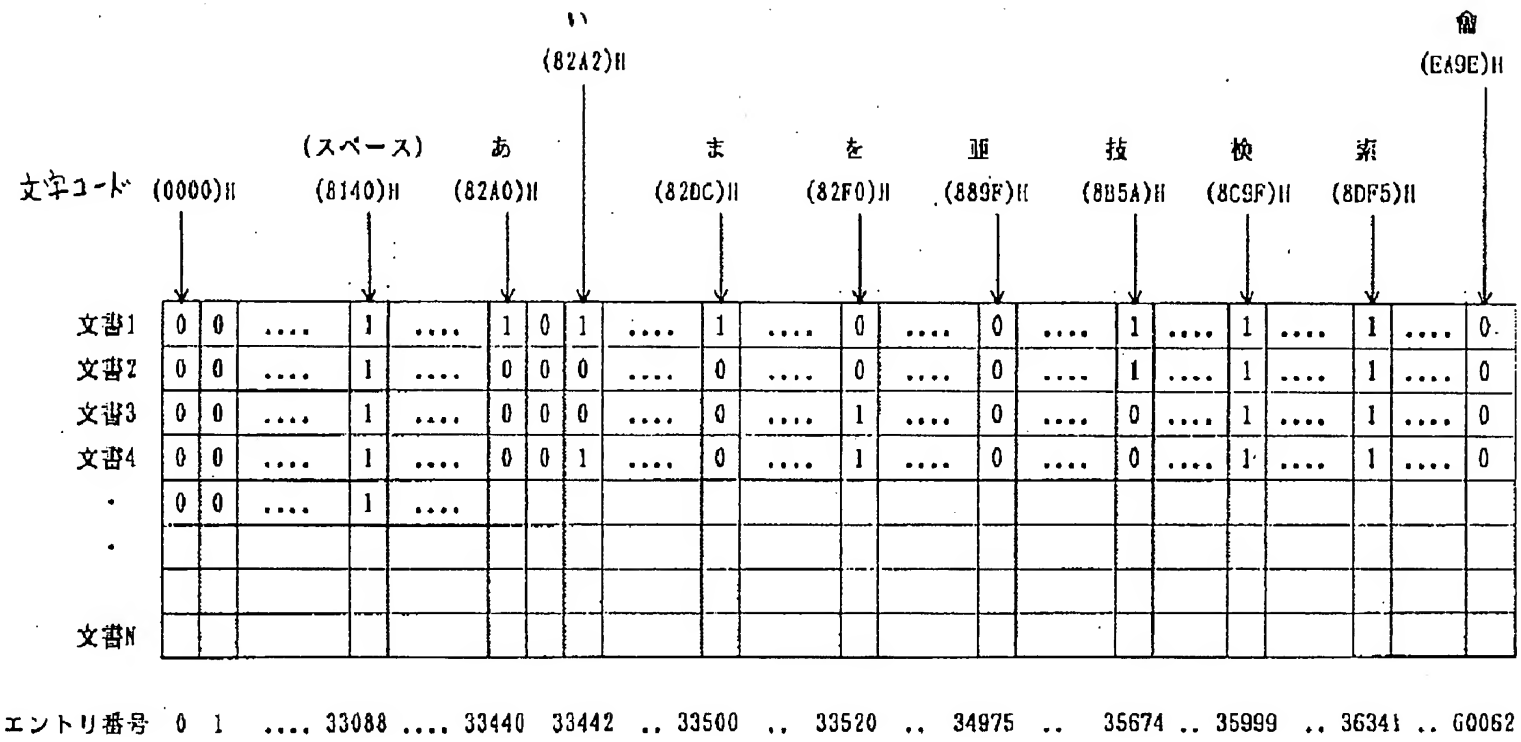
【図７】

〈文字成分表サーチ方法〉



【図６】

〈文字コード依存型文字成分表の構造〉



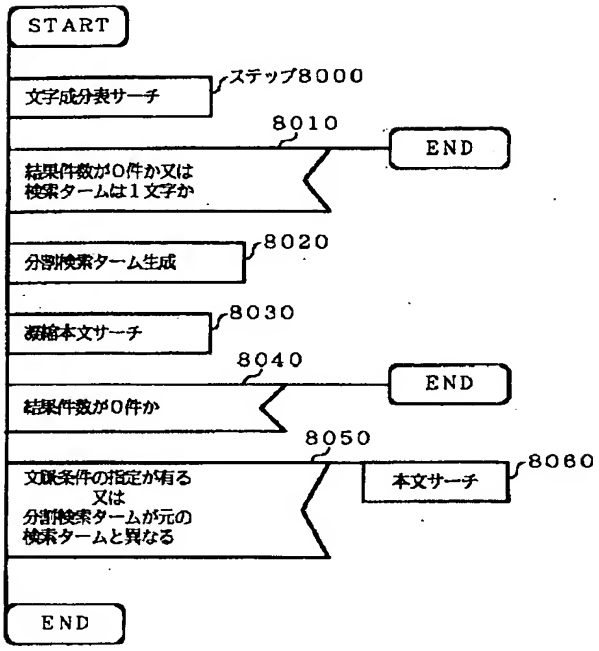
【図６】

特開２００１－３４４２８２

【図８】

【図8】

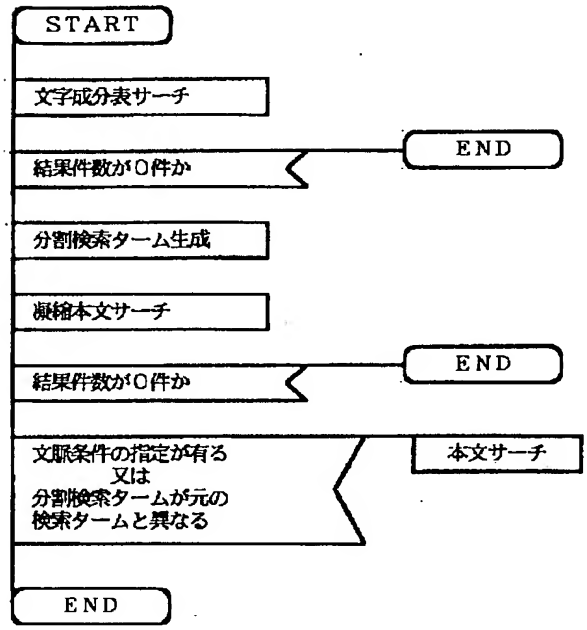
〈階層型プリサーチ手順〉



【図１４】

【図14】

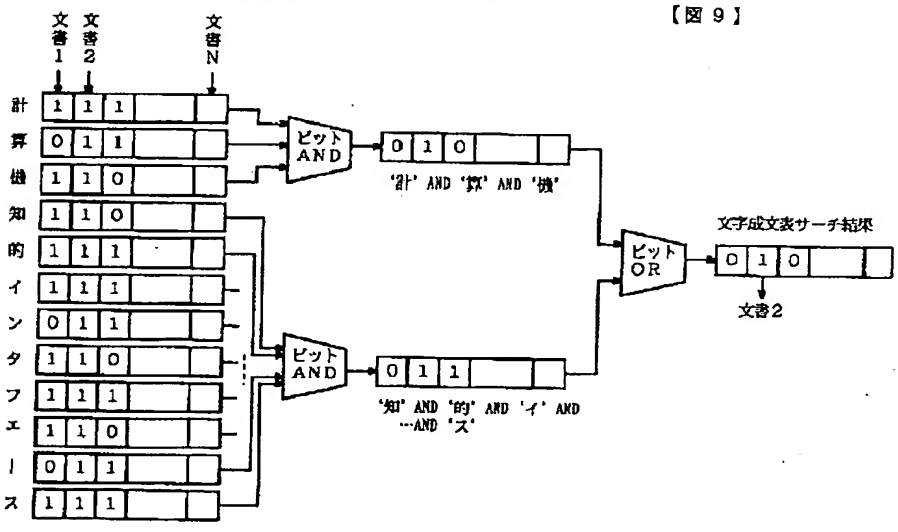
〈ハッシング型文字成分表を用いた階層型プリサーチ手順〉



【図９】

【図９】

〈複数検索タームに対する文字成文表サーチ方法〉



【図１８】

【図１８】

〈ハッシュエントリ変換表〉

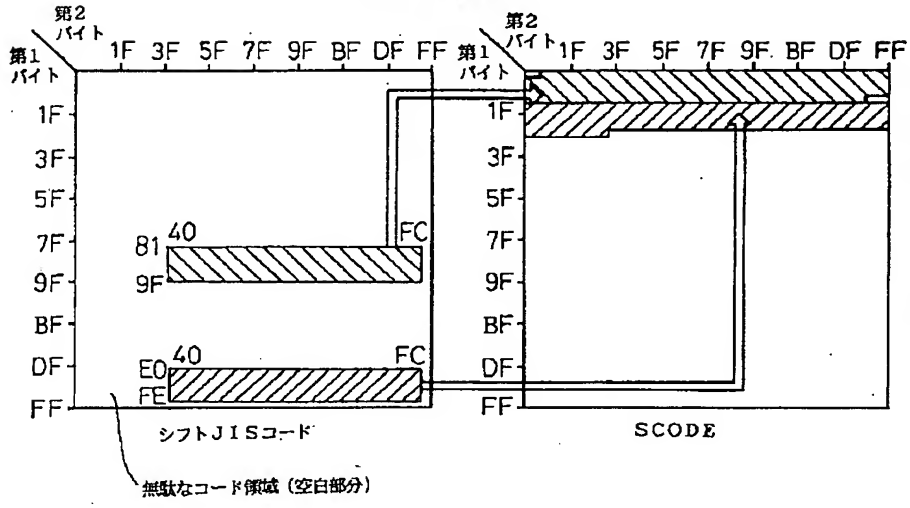
検・・・

SCODE	エントリ番号
・	・
・	・
・	・
(095D)H	150
(095E)H	356
(095F)H	231
(0960)H	483
(0961)H	2
(0962)H	256
(0963)H	25
(0964)H	67
・	・
・	・
・	・

【図１１】

【図１１】

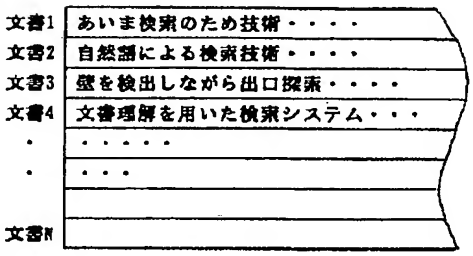
<文字コード変換方法>



【図３３】

【図３３】

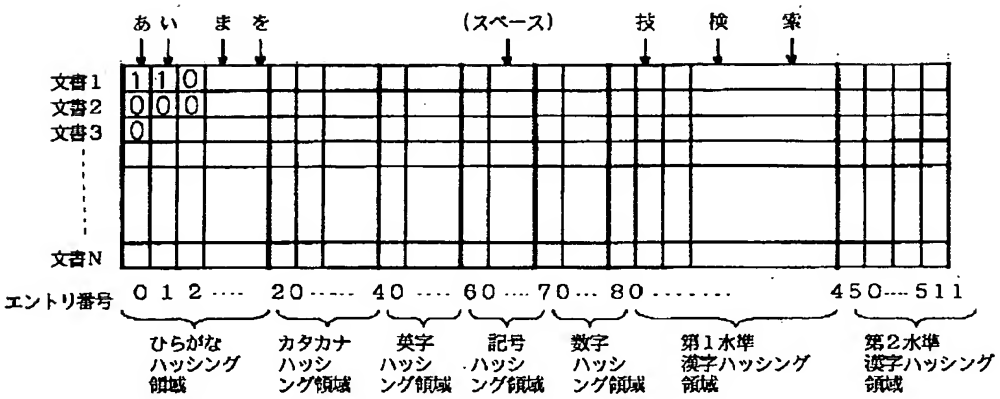
<文字として格納した文字成分表>



【図１５】

【図１５】

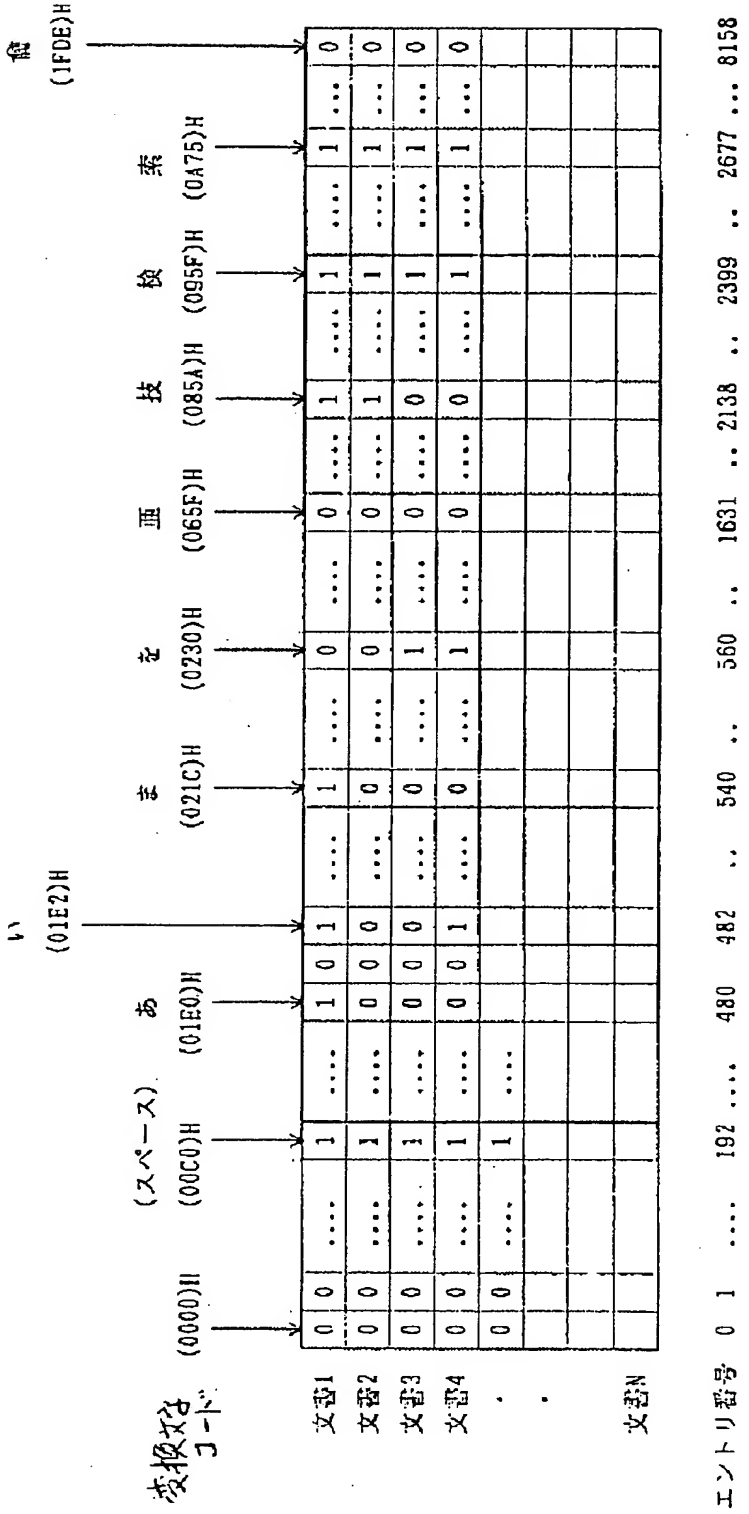
〈文字種を考慮したハッシングによる文字成分表の構造〉



【図１２】

【図１２】

〈文字コード変換を用いた文字成分表の構造〉

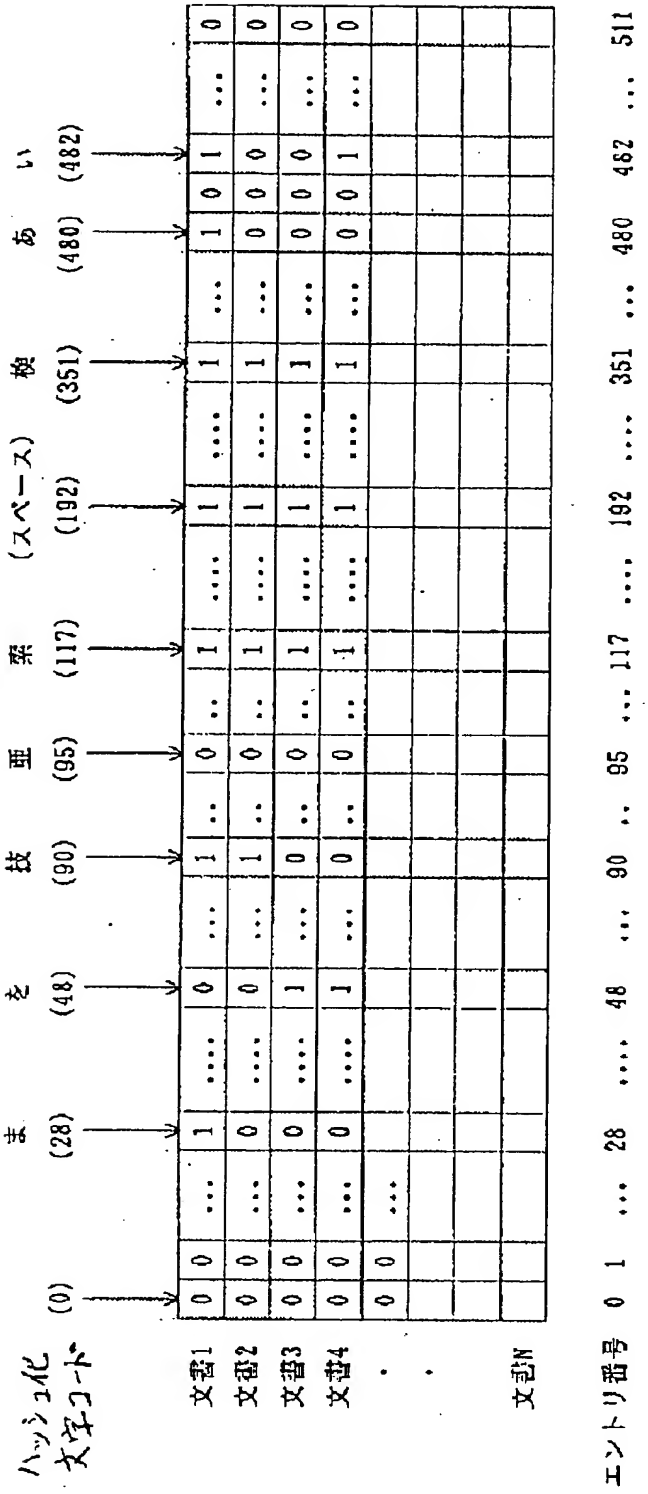


【図１３】

【図１３】

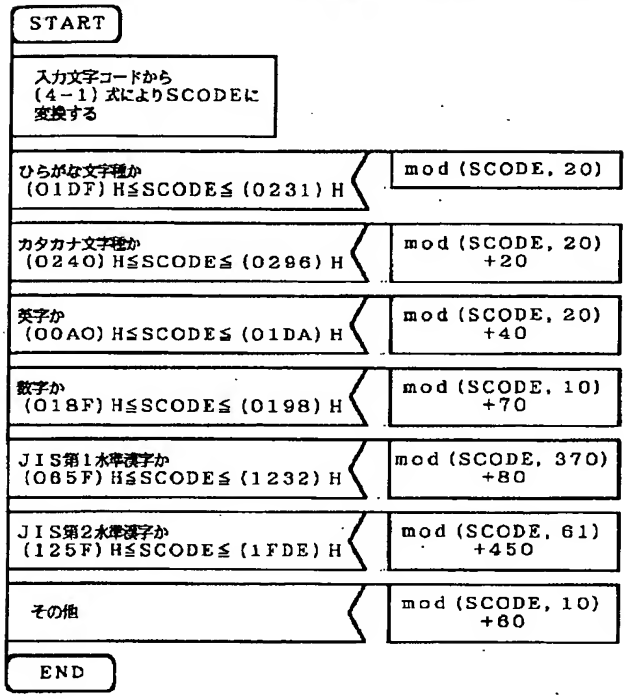
〈ハッシュ化エントリによる文字成分表の構造〉

h(SCODE)=mod(SCODE,512)の場合



【図１６】

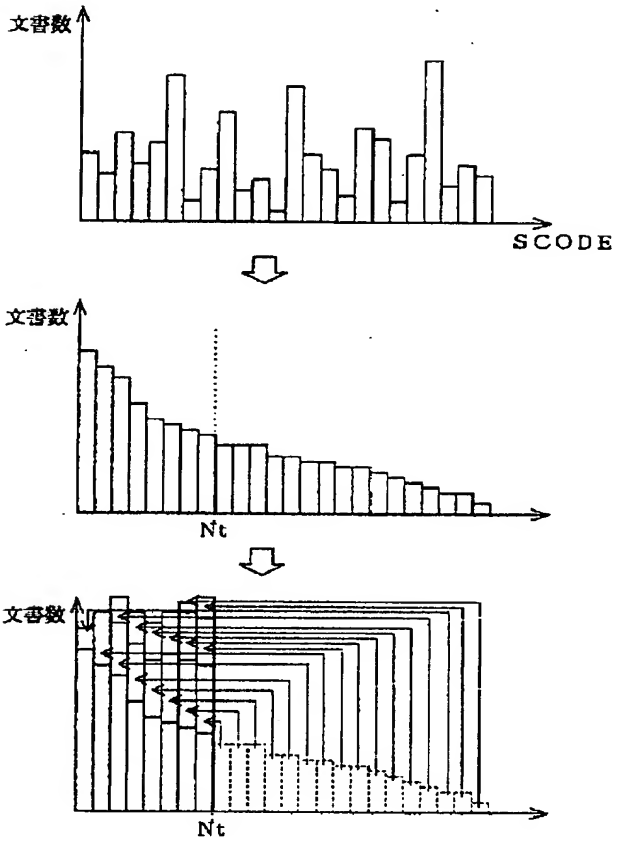
【図16】
〈文字種を考慮したハッシングによる文字成分表を用いた階層型ブリサーチ手順〉



【図１７】

【図１７】

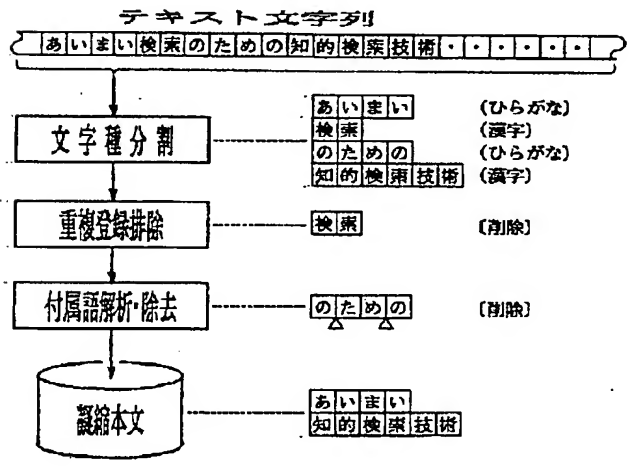
〈頻度情報を考慮したハッシング方法〉



【図１９】

【図１９】

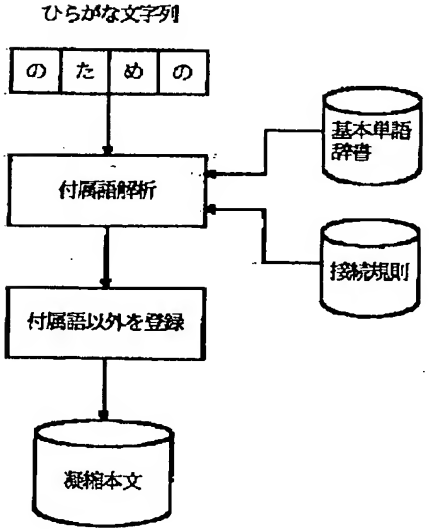
〈文字種分割・重複語排除・付属語除去型凝縮本文の作成方法〉



【図２０】

【図20】

〈付属語除去方法〉



【図２２】

【図２２】

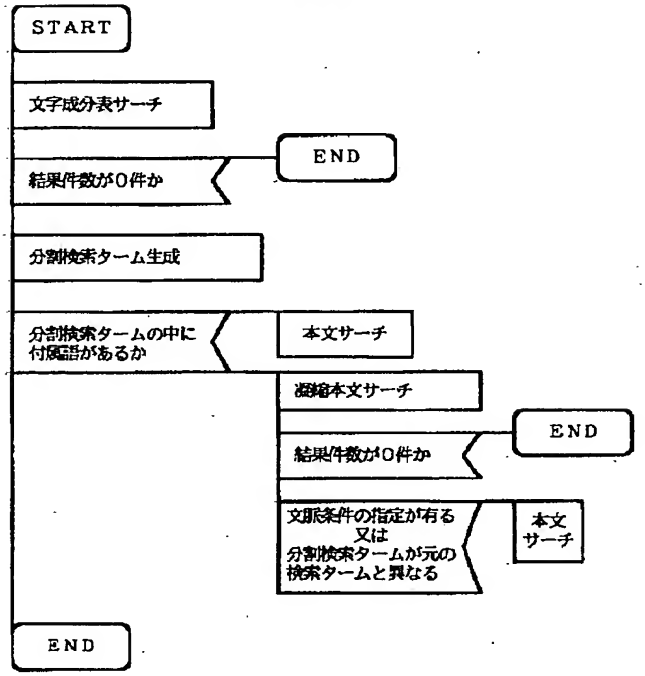
接　続　規　則

接続1 ：＜動詞－もつ連体形＞ ＋ ＜名詞－こと＞
接続2 ：＜動詞－もつ連体形＞ ＋ ＜名詞－ため＞
接続3 ：＜名詞－こと＞ ＋ ＜助詞－が＞
接続5 ：＜動詞－する連体形＞ ＋ ＜名詞－こと＞
接続6 ：＜助詞－の＞ ＋ ＜名詞－ため＞
接続7 ：＜名詞－ため＞ ＋ ＜助詞－の＞
接続8 ：＜接続詞＞ ＋ ＜助詞－は＞
接続9 ：＜動詞－する未然形＞ ＋ ＜助詞－ば＞
接続10：＜名詞－こと＞ ＋ ＜助詞－に＞
接続11：＜助詞－で＞ ＋ ＜動詞－ある＞
接続12：＜動詞－ある未然形＞ ＋ ＜助詞－ば＞
接続13：＜動詞－ある連用形＞ ＋ ＜助詞－た＞

【図２３】

【図２３】

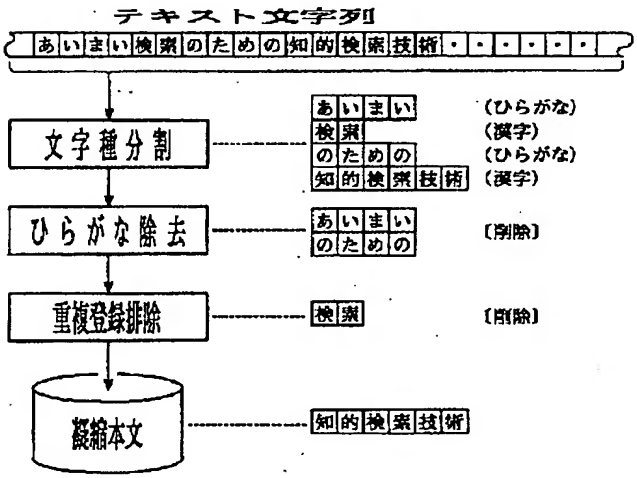
〈文字種分割、重複排除、付属語除去型凝縮本文を用いた階層型プリサーチ手順〉



【図２４】

【図２４】

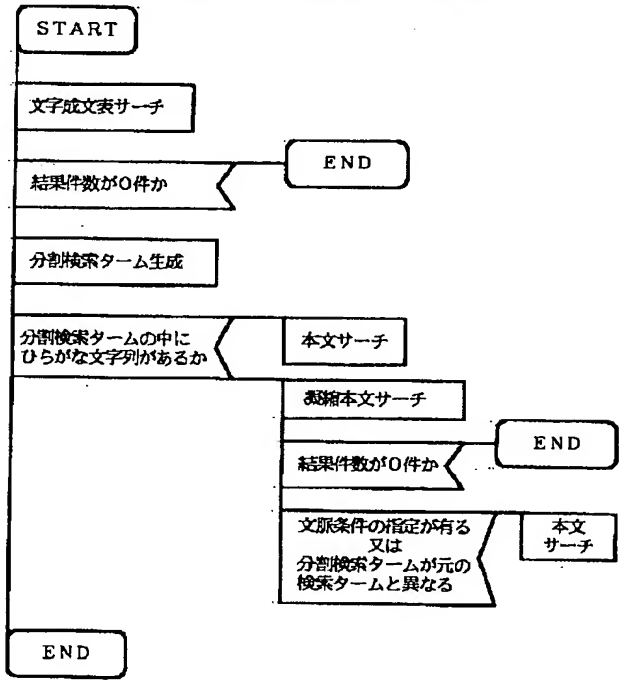
〈文字種分割・重複排除・ひらがな除去型凝縮本文の作成方法〉



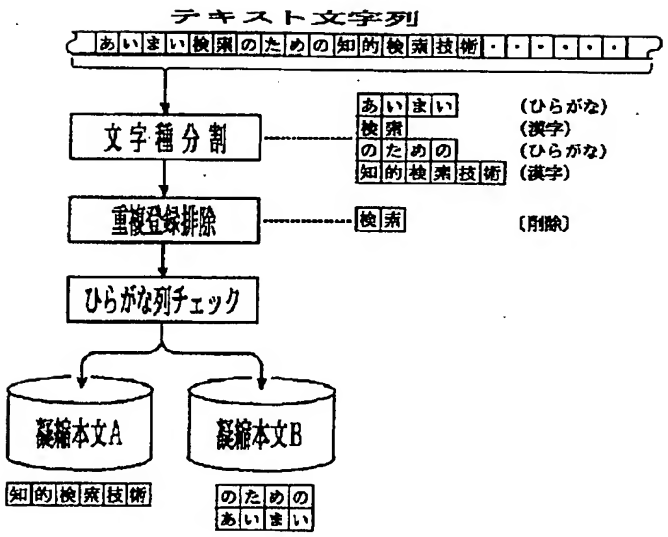
【図２５】

【図２５】

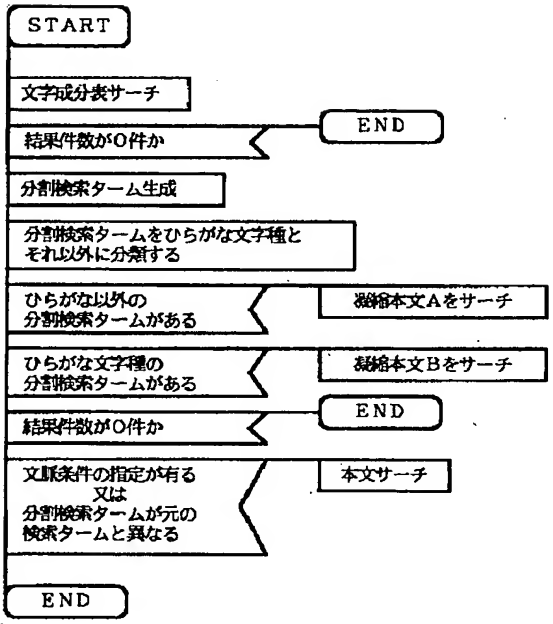
〈文字種分割、重複排除、ひらがな除去型凝縮本文を用いた階層プリサーチ手順〉



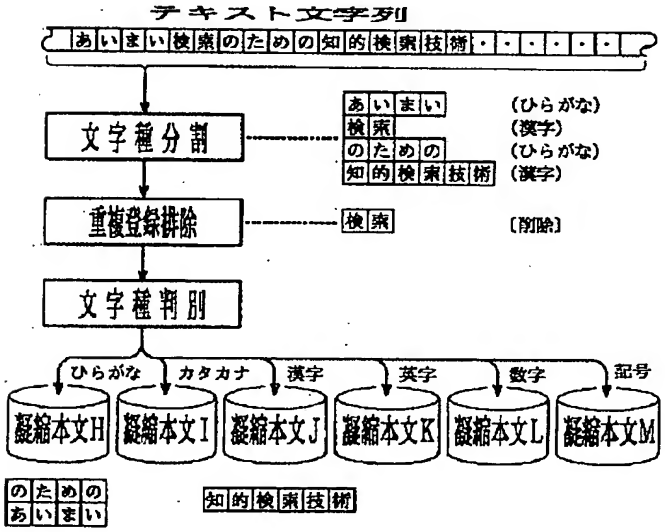
【図２６】



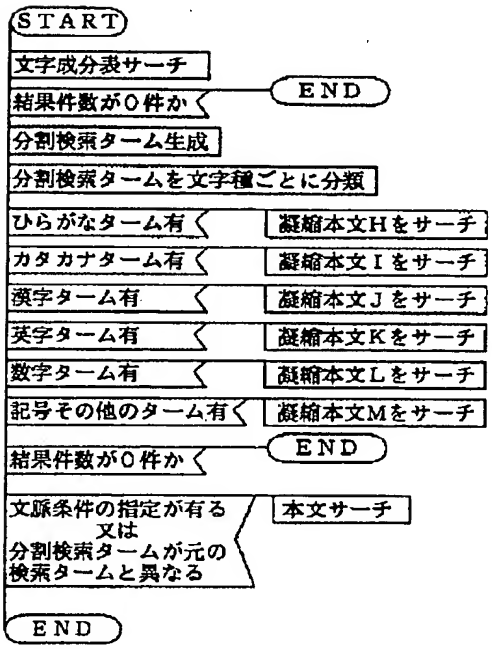
【図２７】



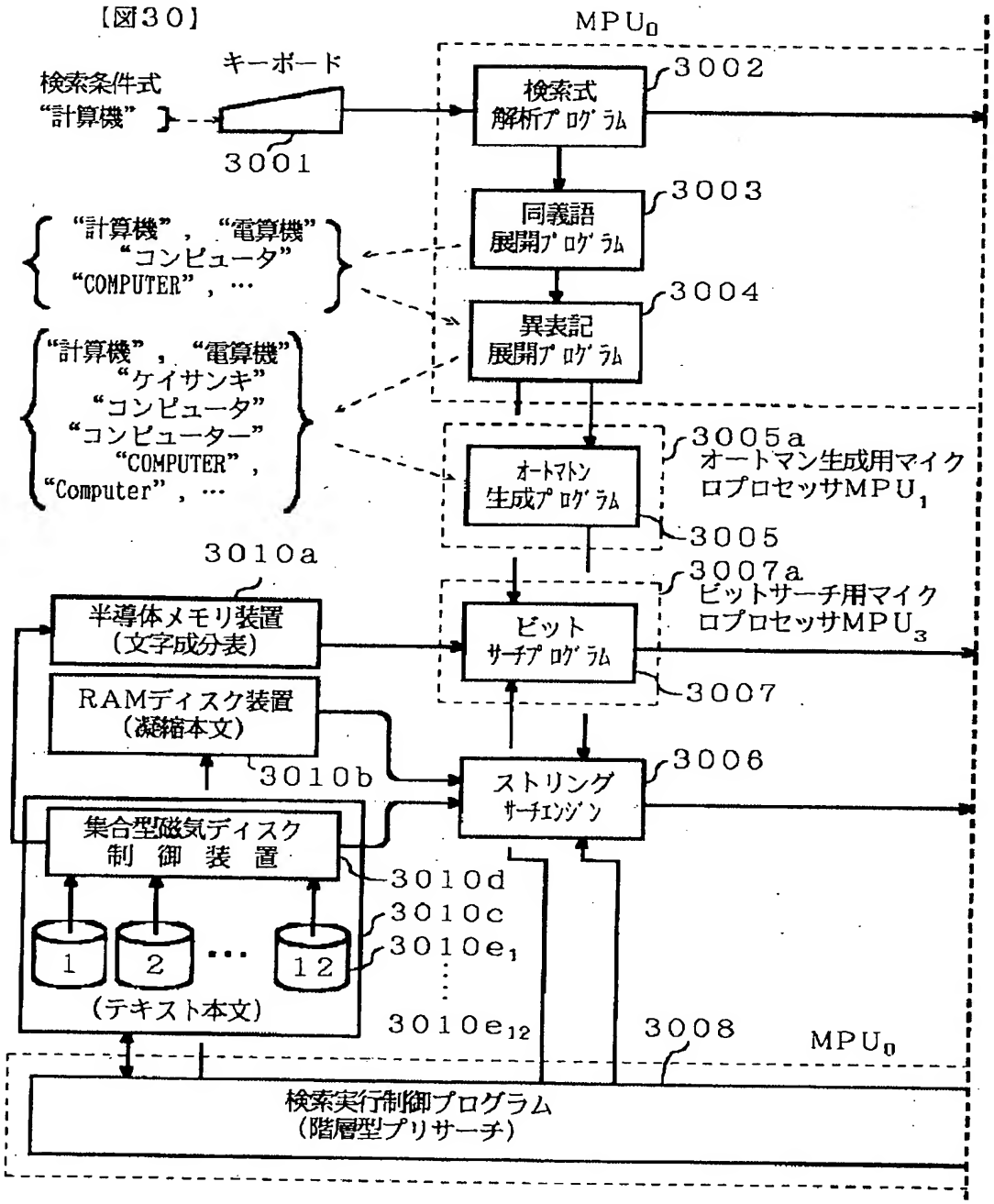
【図２８】



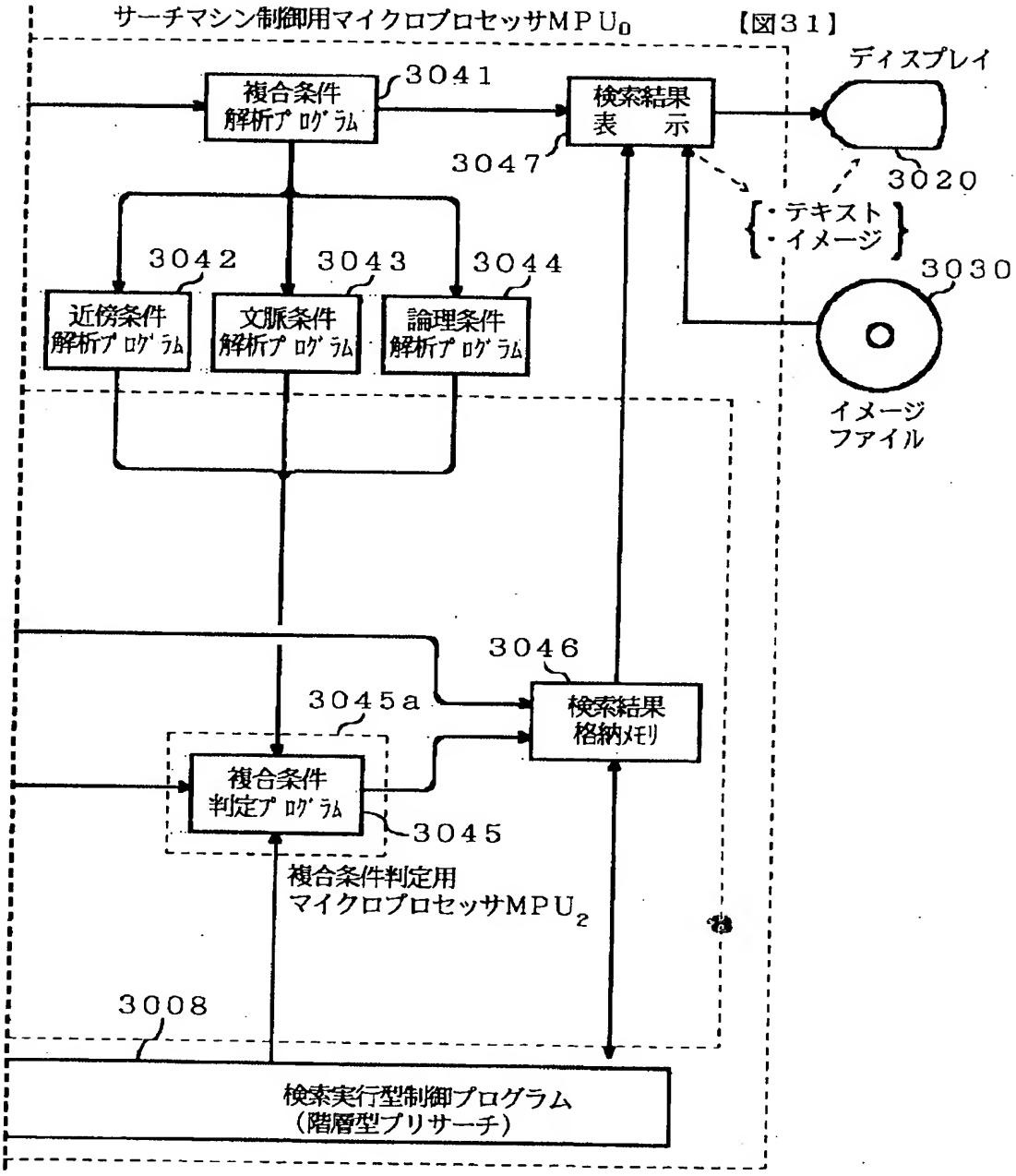
【図２９】



【図３０】

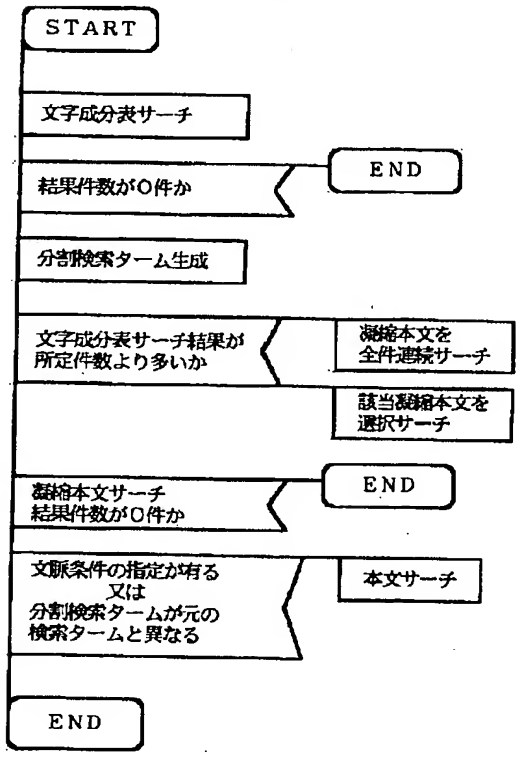

【図30】
MPU0
検索条件式
“計算機”
キーボード
3001
検索式
解析プログラム
3002
同義語
展開プログラム
3003
“計算機”，“電算機”
“コンピュータ”
“COMPUTER”，…
異表記
展開プログラム
3004
“計算機”，“電算機”
“ケイサンキ”
“コンピュータ”
“コンピューター”
“COMPUTER”，
“Computer”，…
オートマトン
生成プログラム
3005a
オートマン生成用マイクロプロセッサMPU1
3005
3010a
半導体メモリ装置
（文字成分表）
ビット
サーチプログラム
3007a
ビットサーチ用マイクロプロセッサMPU3
3007
RAMディスク装置
（凝縮本文）
3010b
ストリング
サーチエンジン
3006
集合型磁気ディスク
制御装置
3010d
3010c
3010e1
1
2
12
（テキスト本文）
3010e12
3008
MPU0
検索実行制御プログラム
（階層型プリサーチ）

【図３１】


サーチマシン制御用マイクロプロセッサMPU0
【図31】
ディスプレイ
複合条件解析プログラム
3041
検索結果表示
3047
3020
テキスト
イメージ
3030
3042
3043
3044
近傍条件解析プログラム
文脈条件解析プログラム
論理条件解析プログラム
イメージファイル
3046
3045a
検索結果格納メモリ
複合条件判定プログラム
3045
複合条件判定用マイクロプロセッサMPU2
3008
検索実行型制御プログラム（階層型プリサーチ）

【図３２】

【図32】
〈凝縮本文を磁気ディスクへ格納した場合の
階層ブリサーチ手順〉



フロントページの続き

(72)発明者　加藤　寛次
東京都国分寺市東恋ケ窪１丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　川口　久光
東京都国分寺市東恋ケ窪１丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　嶺岸　直材
神奈川県川崎市幸区鹿島田890番地の12
株式会社日立製作所情報システム工場内
Ｆターム(参考)　5B075 ND03 NR02 NR14 NS10 PP02
PQ02 QM01